# 実践的なHDR出力対応
## ～レンダリングパイプラインの構築～

シリコンスタジオ株式会社
ミドルウェア開発部
リード・ソフトウェアエンジニア
川瀬 正樹

Silicon Studio
©Silicon Studio Corp., all rights reserved!

# 内容

- 本セッションの目的
- 色空間
- 前提として押さえておきたいこと
- 基本的なレンダリングフローの検討
- その他の考慮すべきこと
- 互換性重視レンダリングフローの構築
- まとめ


©Silicon Studio Corp., all rights reserved.

# 本セッションの目的

本セッションの目的
色空間
前提として押さえておきたいこと
基本的なレンダリングフローの検討
その他の考慮すべきこと
互換性重視レンダリングフローの構築
まとめ

Silicon Studio ©Silicon Studio Corp. all rights reserved!

# 本セッションの目的

- HDR出力レンダリングフロー構築のためのノウハウ
  - レンダリングフローを構築する際の考え方
  - SDRとの互換性の保ち方
- レンダリングフローを独自に構築できる
  - 必要な条件によって最適なフローも変化
  - 各々に最適なフローを必要に応じて構築できる


©Silicon Studio Corp., all rights reserved.

# 色空間

本セッションの目的
色空間
前提として押さえておきたいこと
基本的なレンダリングフローの検討
その他の考慮すべきこと
互換性重視レンダリングフローの構築
まとめ

Silicon Studio ©Silicon Studio Corp. all rights reserved!

# 色空間(Color Space)とは？

- 色を定量化するための空間
  - よく知られる色空間(色空間として参照される規格等)

    sRGB, Adobe RGB, ProPhoto RGB, DCI-P3, ACEScc, ACEScg,
    BT.709, BT.2020, BT.2100 PQ, BT.2100 HLG, etc.

- 色域・伝達関数・輝度や鑑賞条件等で定義される
  - 色空間がどのような定義を含むかは各々異なる
    - 規格によって厳密さや内容も異なる

- 色空間以外の規定も含む規格も
  - 例えばBT.709はビデオ信号規格
    - 色空間だけでなく解像度や走査線、画像伝送レート等も含む


©Silicon Studio Corp., all rights reserved.

# 色空間を構成する要素

- 色域(Gamut)
- 伝達関数(Transfer Function)
- 輝度条件や鑑賞条件、ビット深度等

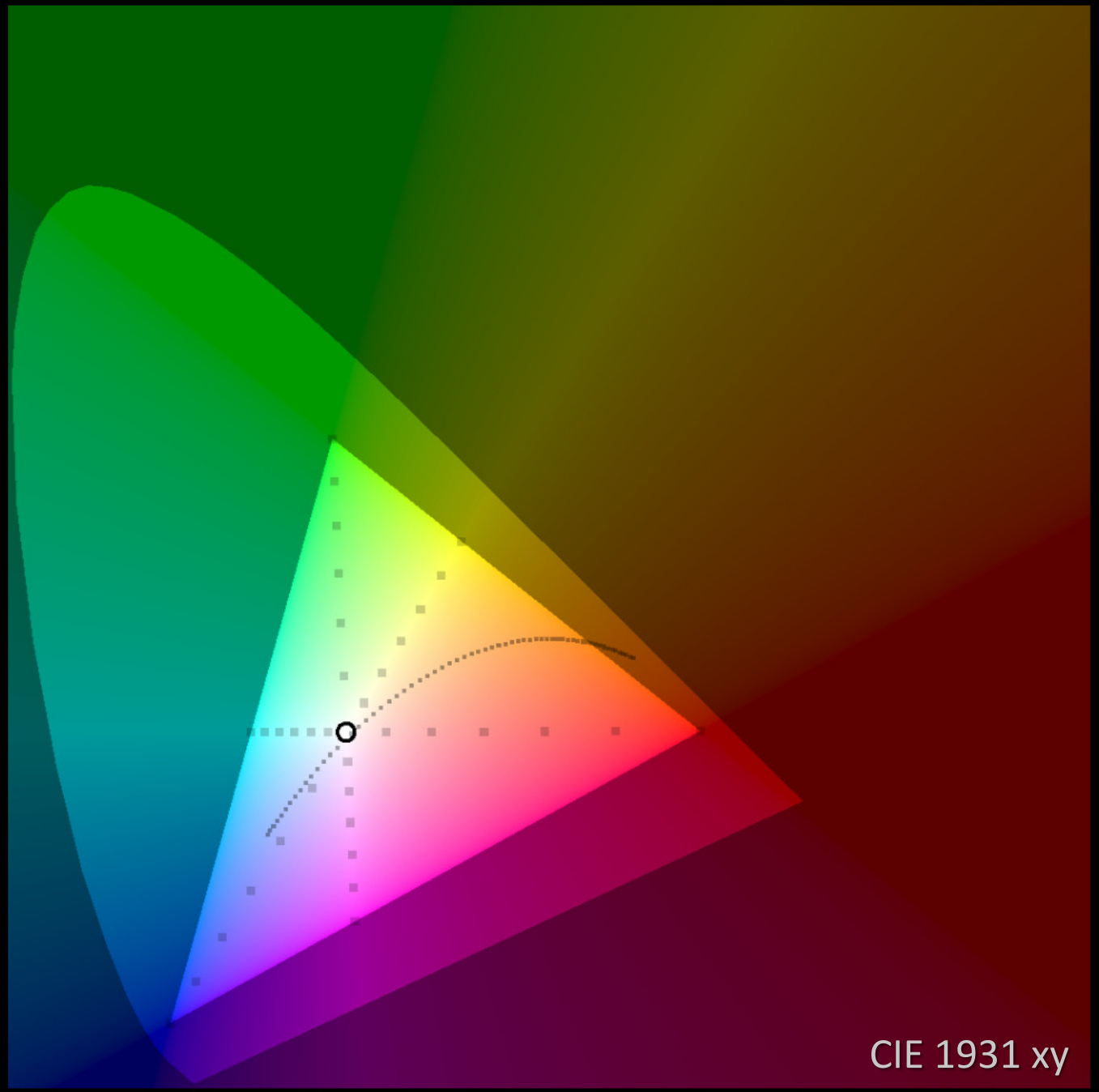

©Silicon Studio Corp., all rights reserved.

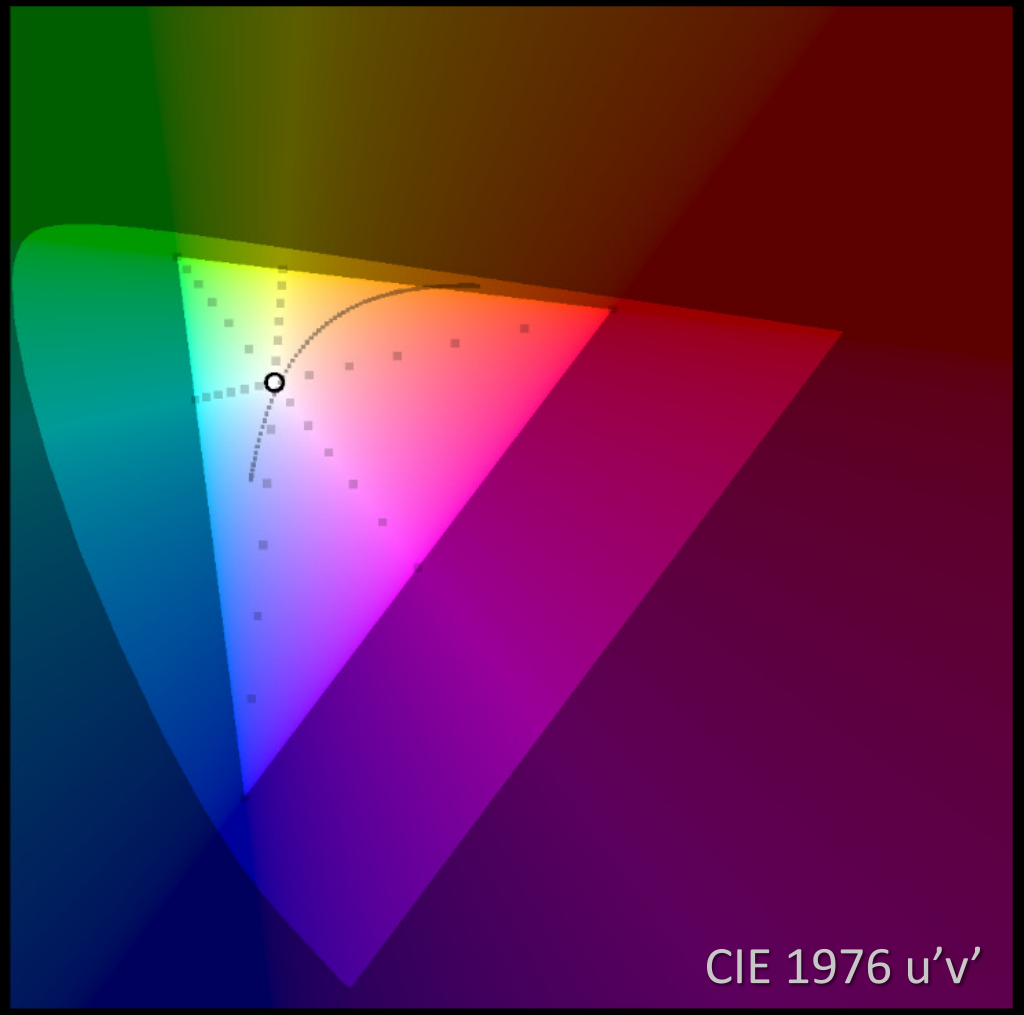
# 色域(Gamut)

- そもそも三原色と白が具体的に何を指しているか？
  - 赤(r,g,b)=(1,0,0)とは物理的にはどのような色？
  - 白(r,g,b)=(1,1,1)とは物理的にはどのような色？
- 三原色と白色点の色度座標で定義される
  - 一般的にCIE 1931 xy色度座標
  - 白色点も定義されるため、色温度も意識する必要あり

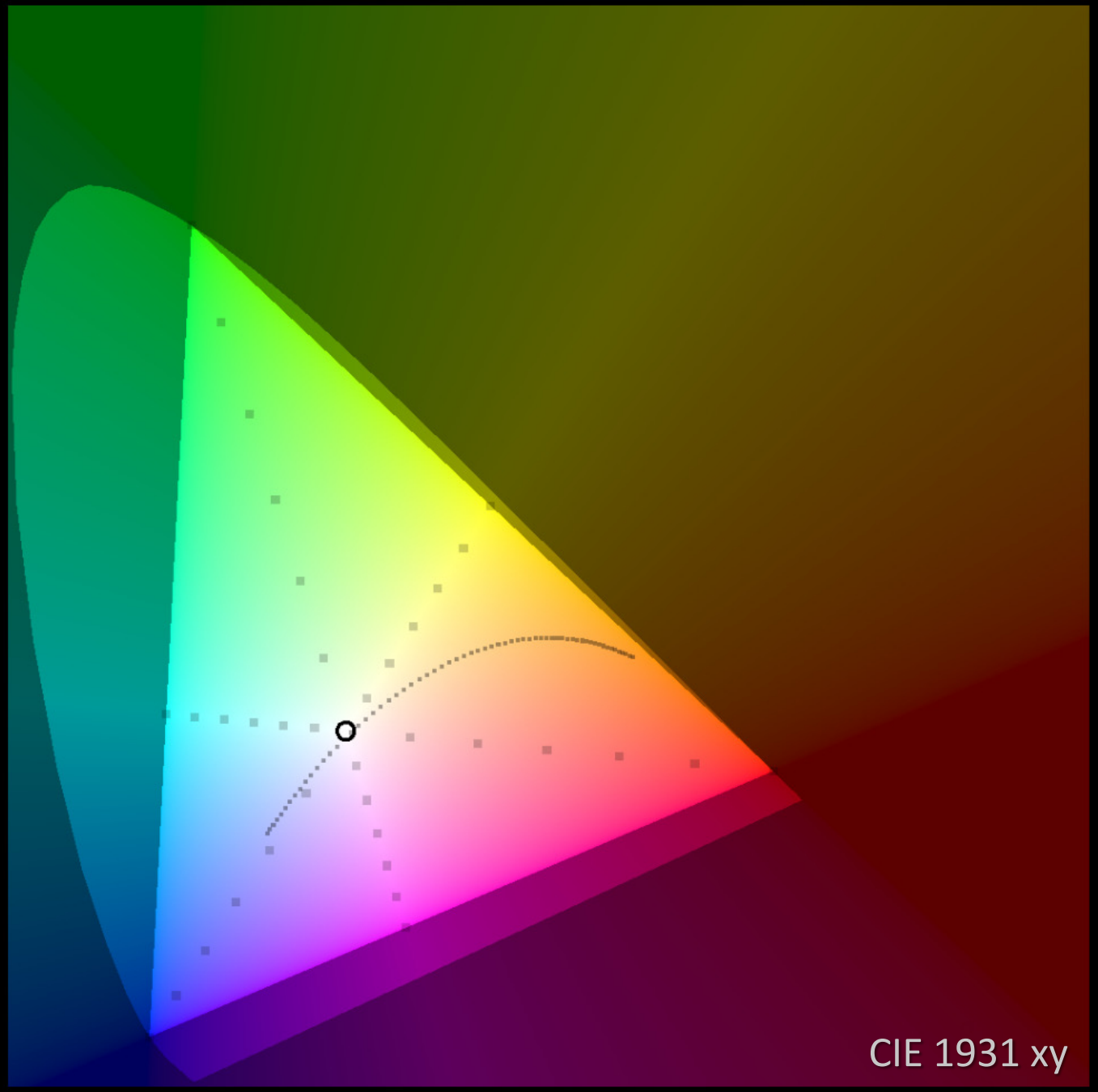

©Silicon Studio Corp., all rights reserved.

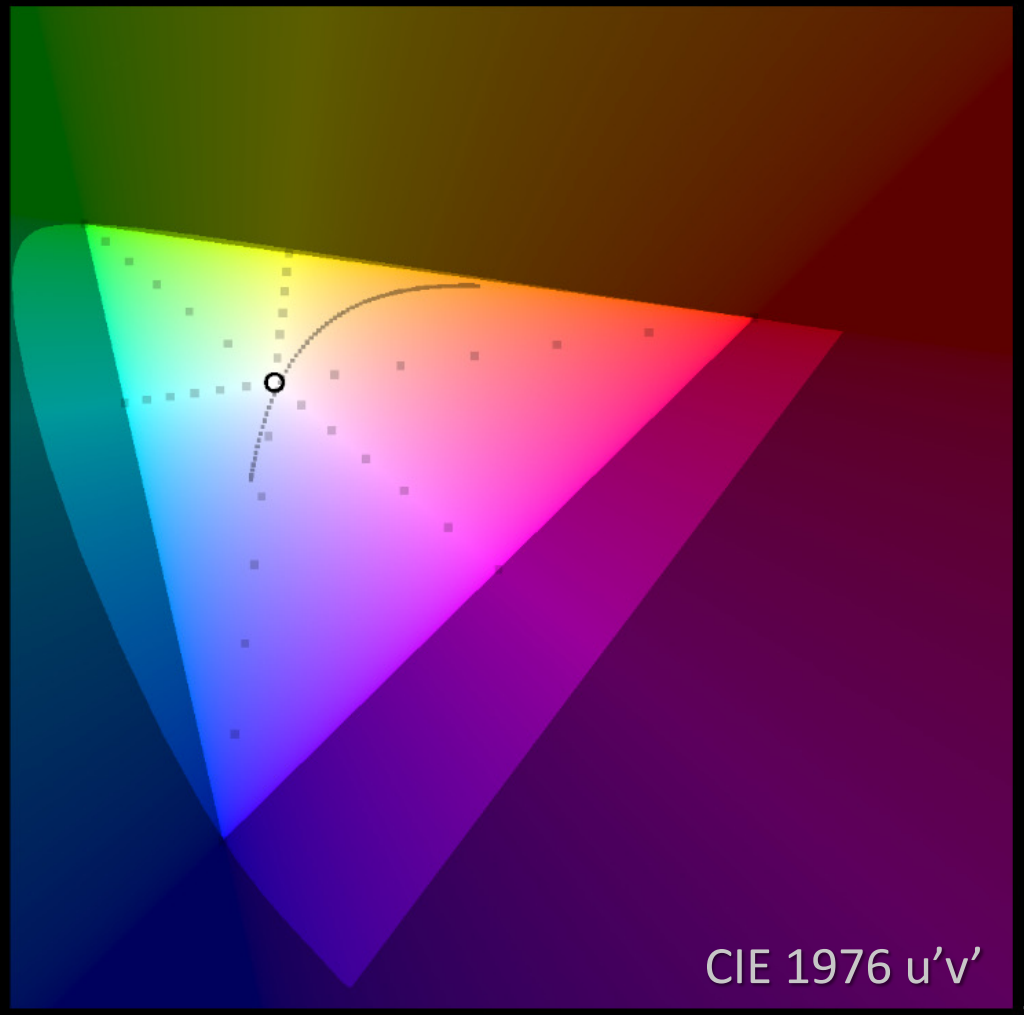
# ITU-R BT.709



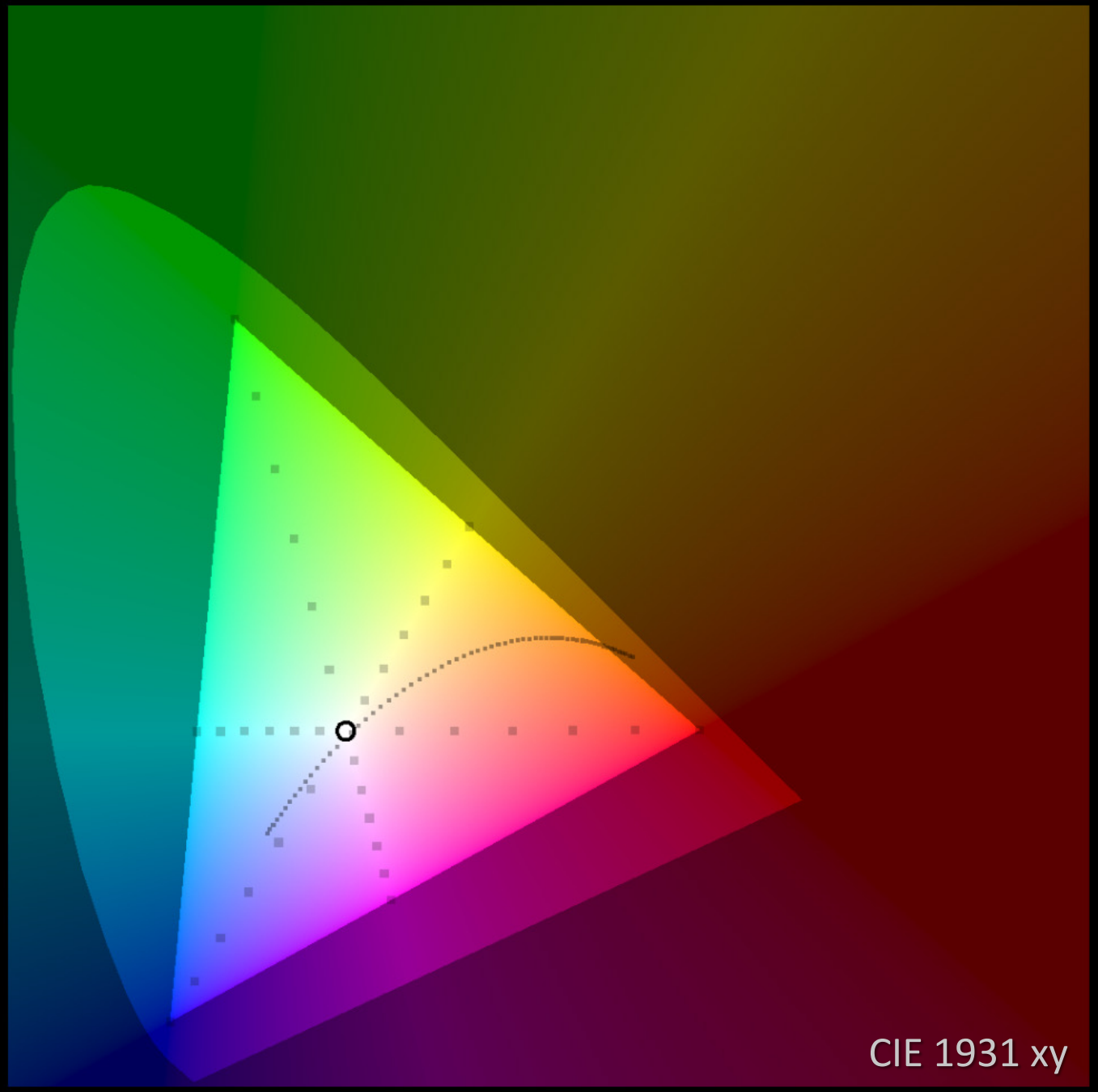



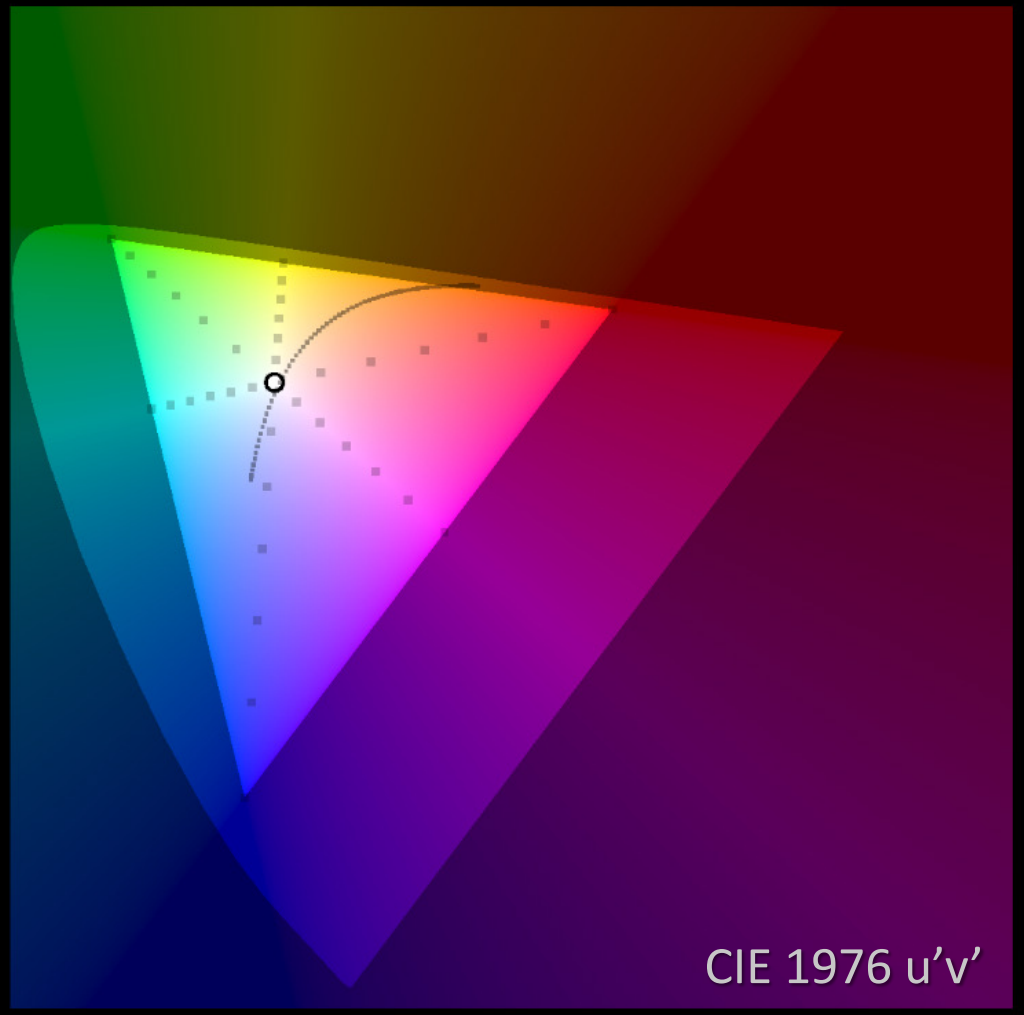
# ITU-R BT.2020



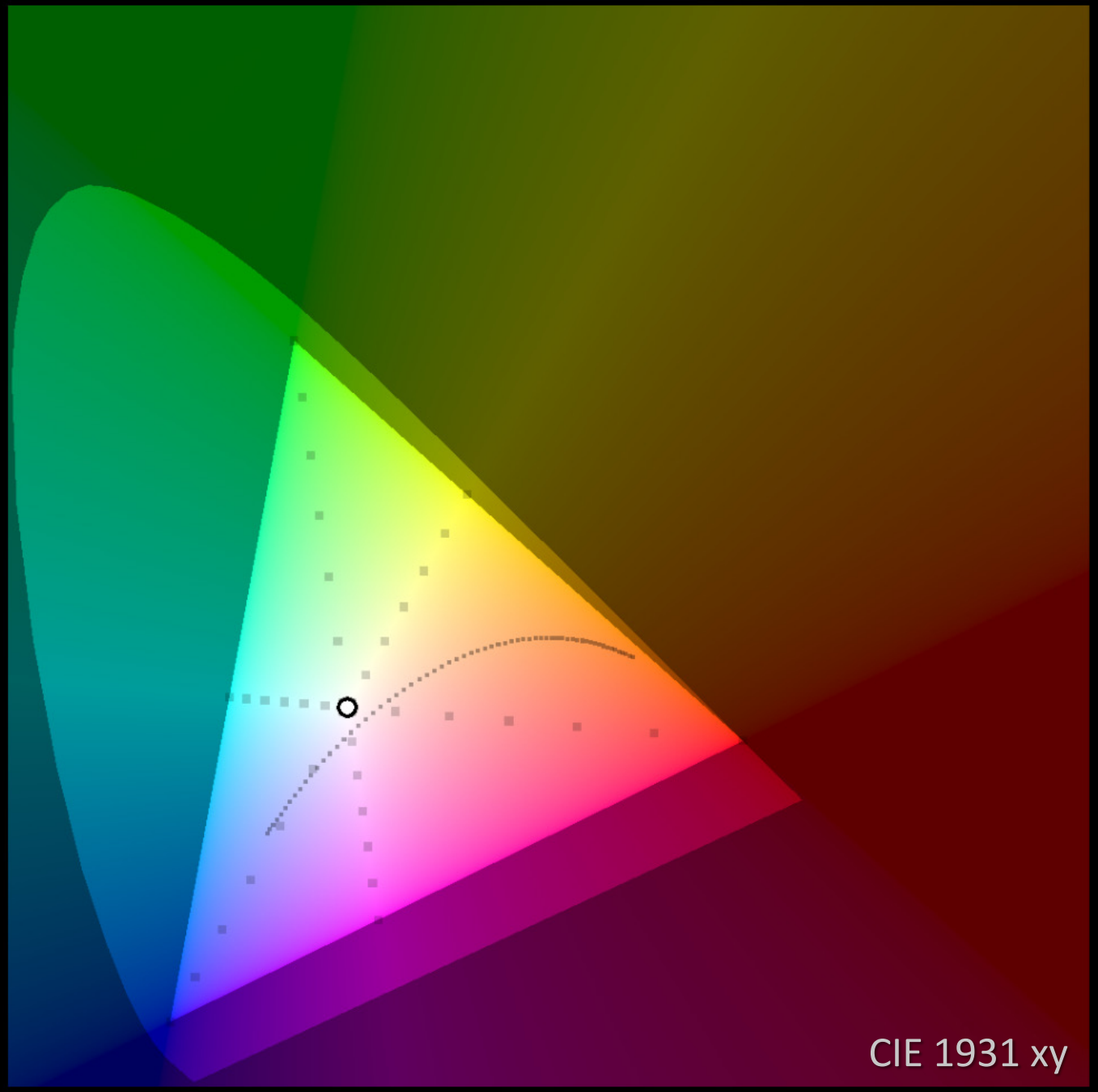



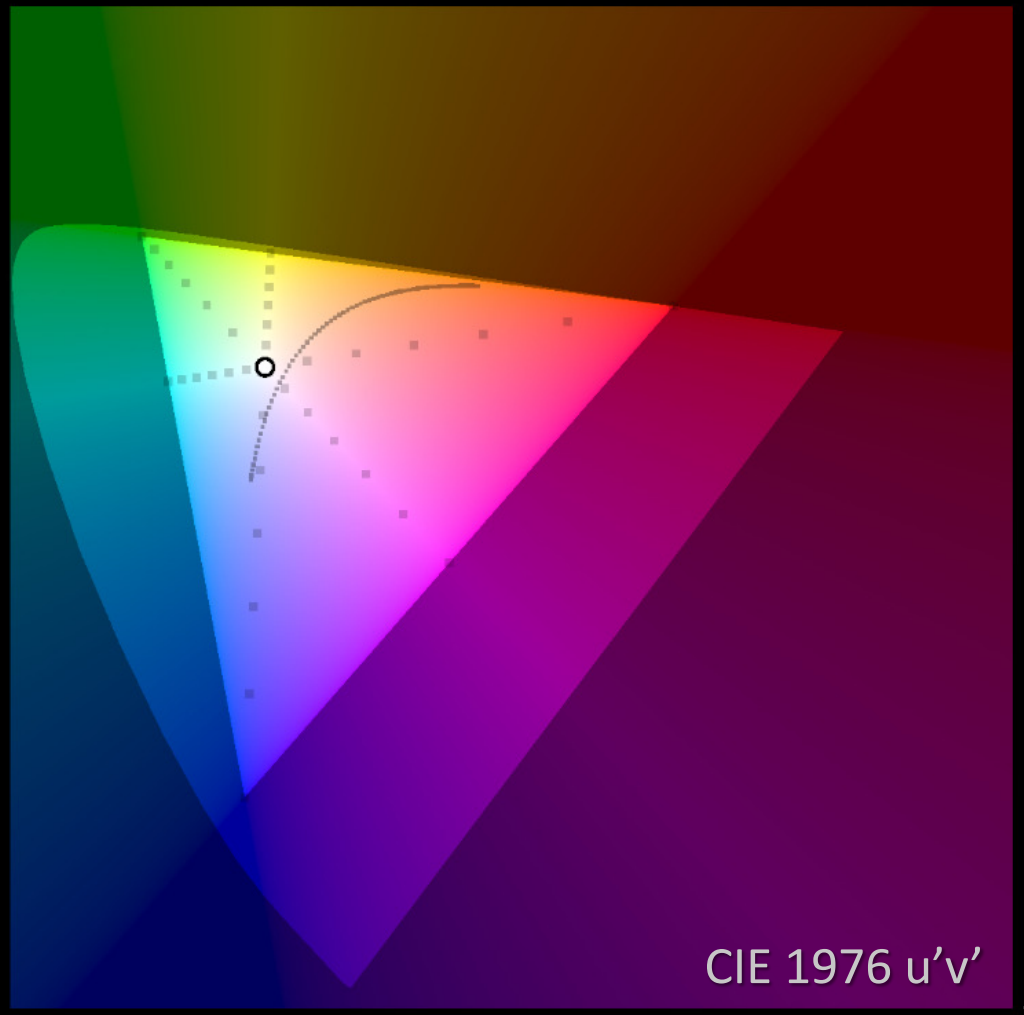
# Adobe RGB



# DCI-P3



# 色空間を構成する要素

- 色域(Gamut)
- 伝達関数(Transfer Function)
- 輝度条件や鑑賞条件、ビット深度等

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 伝達関数(Transfer Function)

- いわゆるガンマカーブ
- EOTFまたはOETFで定義される
  - EOTF(Electro-Optical Transfer Function: 電気－光伝達関数)
    - テレビ側デガンマ
  - OETF(Opto-Electronic Transfer Function: 光－電気伝達関数)
    - コンテンツ側ガンマ補正
- sRGBやAdobe RGB, BT.709等も全て伝達関数をもつ
  - 色域だけを指す訳ではない

Silicon Studio ©Silicon Studio Corp., all rights reserved.

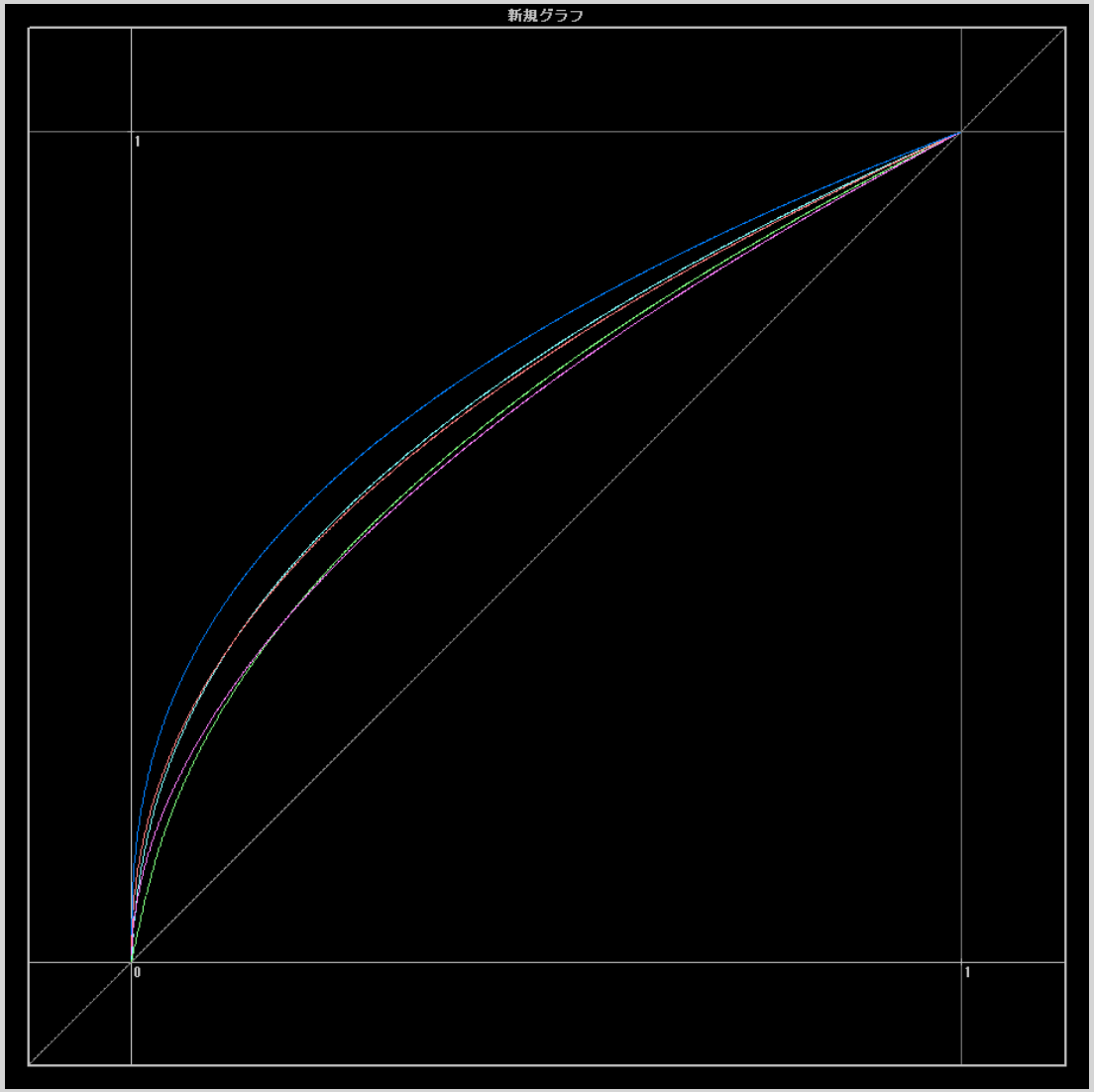
# さまざまな色空間の伝達関数

- 伝送関数(Transfer Function)の例
  - sRGB色空間: 暗部に線形領域をもつガンマカーブ(γ~2.2)
  - BT.709: 暗部に線形領域をもつガンマカーブ(γ~1.9)
    - BT.2020もほぼ同じ(ビット深度によって高精度な定義あり)
  - Adobe RGBやWide Gamut RGB: γ~2.2(563/256=2.19921875)
  - Display P3: sRGBガンマの数値計算精度を上げたもの(γ~2.2)
  - DCI-P3: γ2.6
  - BT.2100 PQ: Perceptual Quantizer / ST.2084
  - BT.2100 HLG: Hybrid Log-Gamma / ARIB STD-B67


©Silicon Studio Corp., all rights reserved.

# OETF例

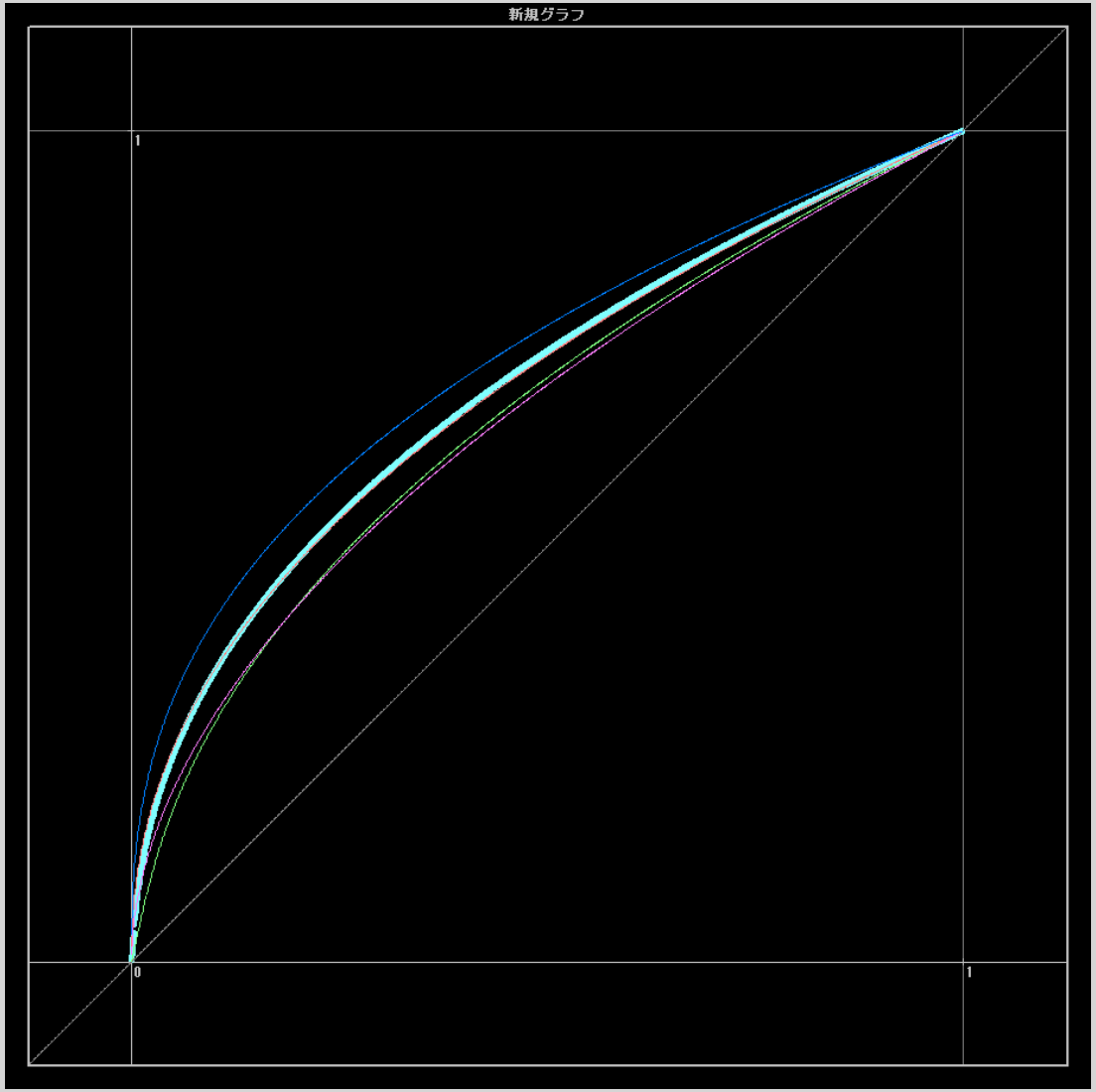
- 代表的なSDR伝達関数
  - sRGB
  - Adobe RGB
  - BT.709 / BT.2020
  - DCI-P3




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OETF例

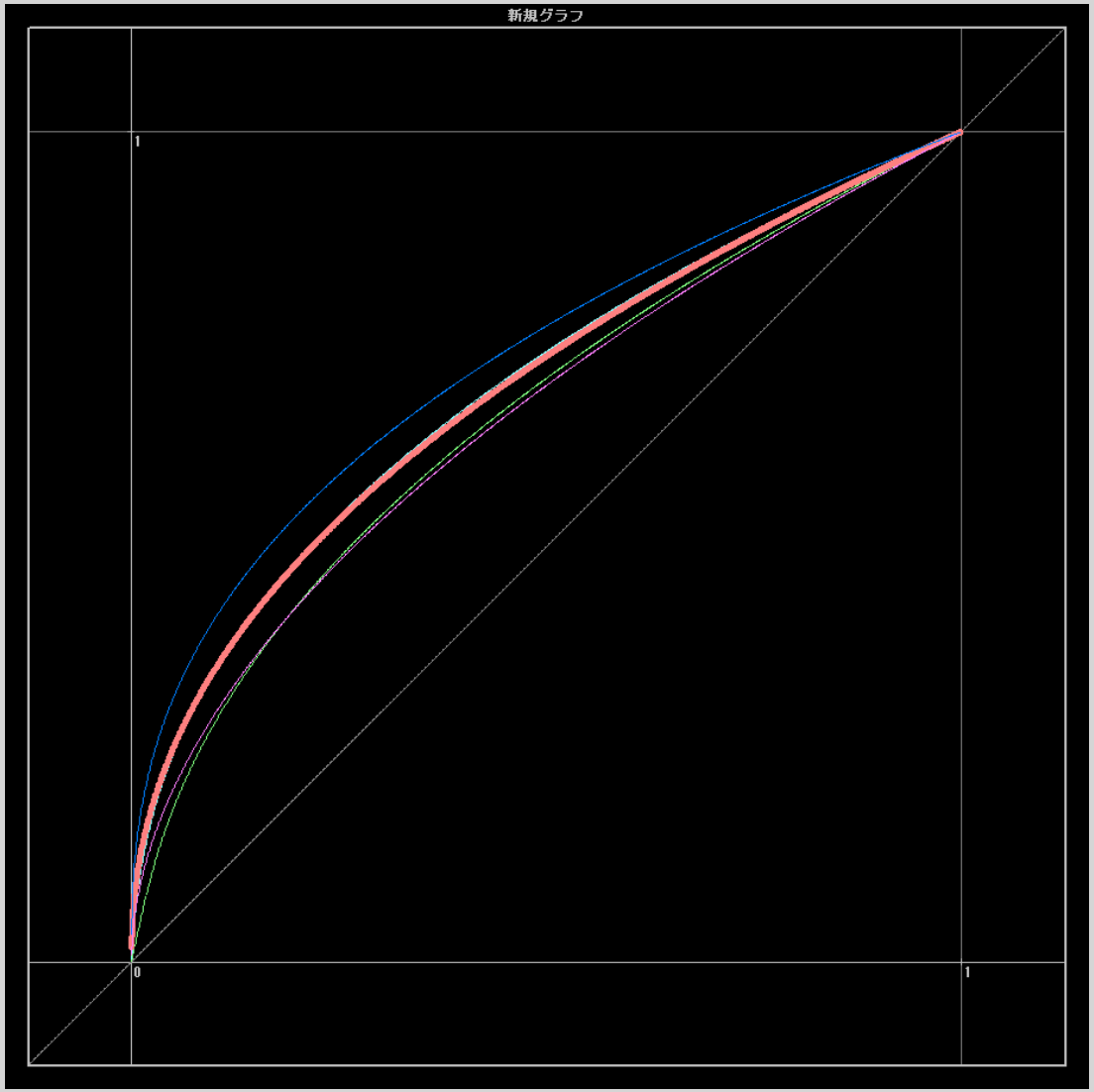
- sRGB
  - $L \leq 0.0031308$
    - $V = 12.92L$
  - $L > 0.0031308$
    - $V = (1 + 0.055)L^{1/2.4} - 0.055$




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OETF例

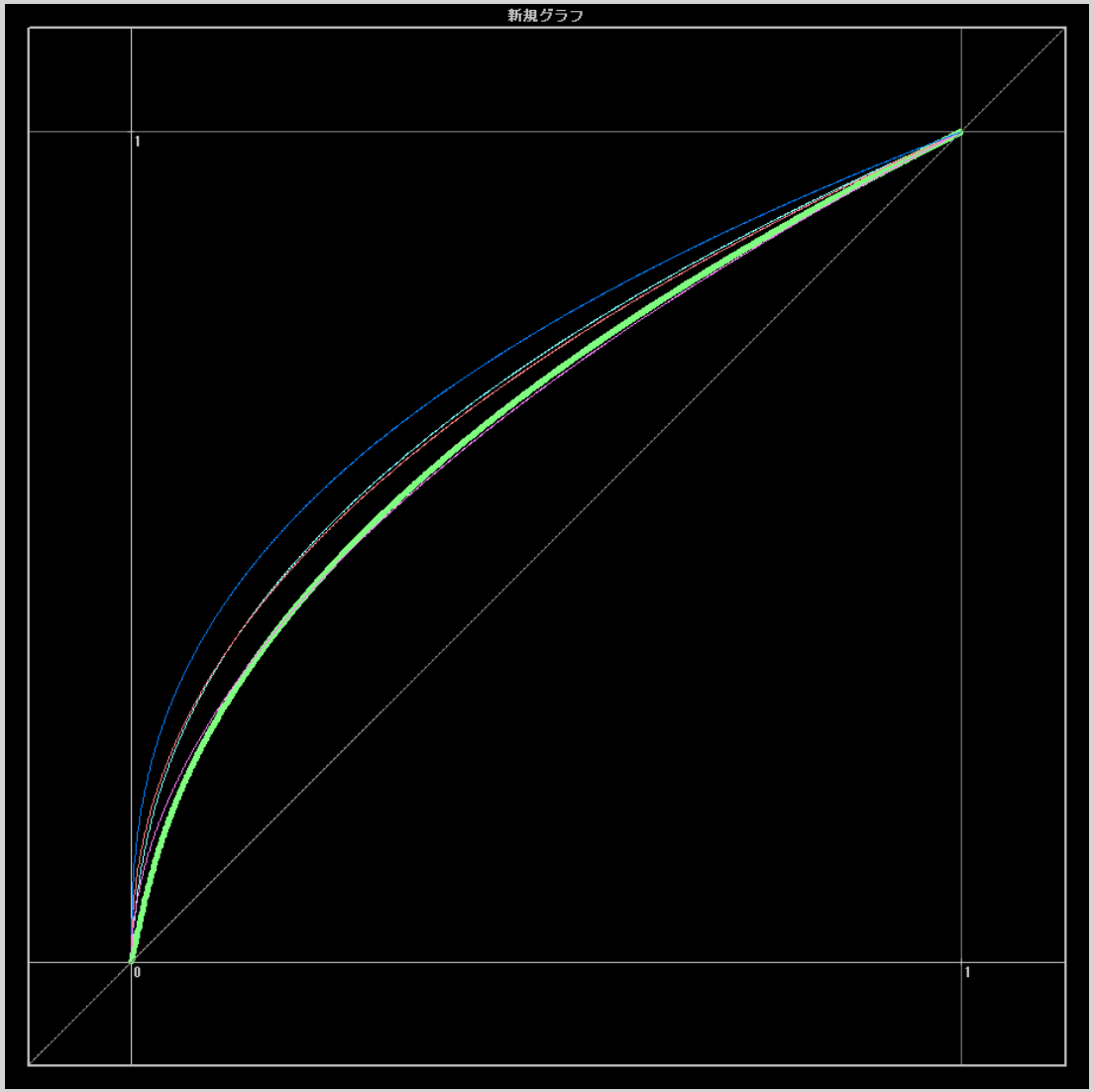
- Adobe RGB
  - $V = L^{256/563} = L^{1/2.19921875}$




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OETF例

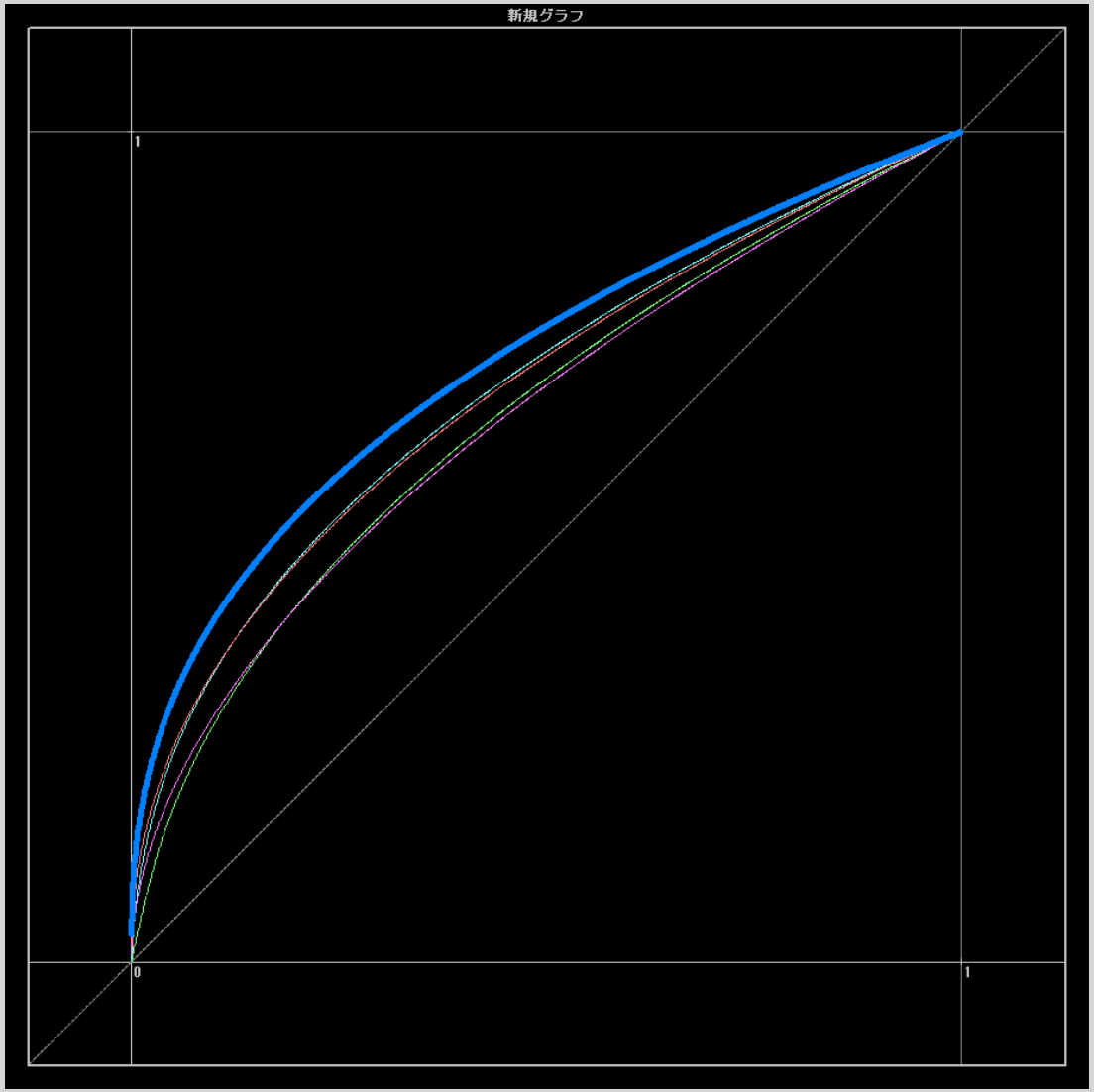
- BT.709 / BT.2020(10-bit RGB)
  - $L \leq 0.018$
    - $V = 4.500L$
  - $L > 0.018$
    - $V = (1 + 0.099)L^{0.45} - 0.099$
- BT.2020(12-bit RGB)
  - $L \leq 0.0181$
    - $V = 4.500L$
  - $L > 0.0181$
    - $V = (1 + 0.0993)L^{0.45} - 0.0993$




Silicon Studio ©Silicon Studio Corp., all rights reserved.

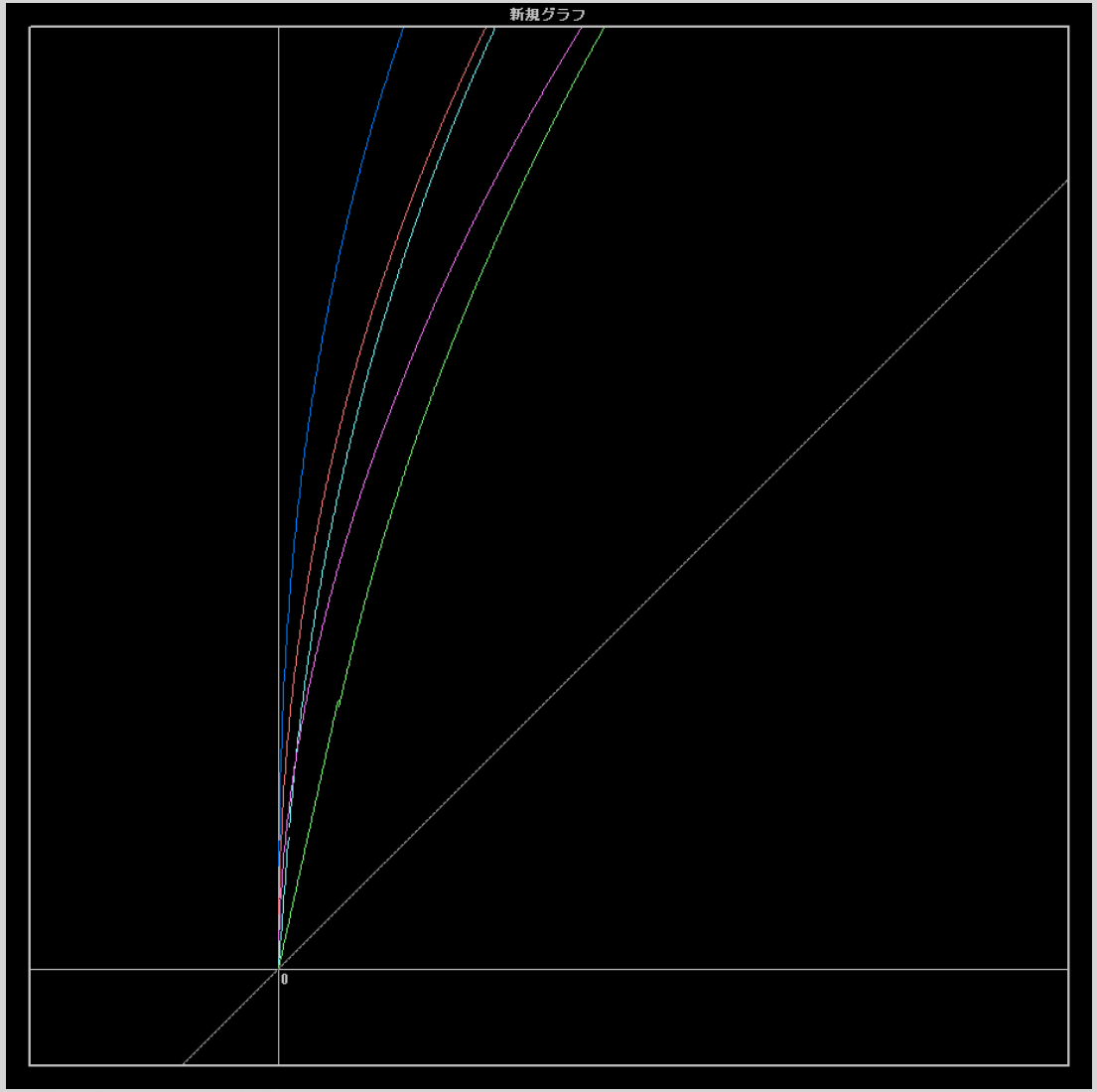
# OETF例

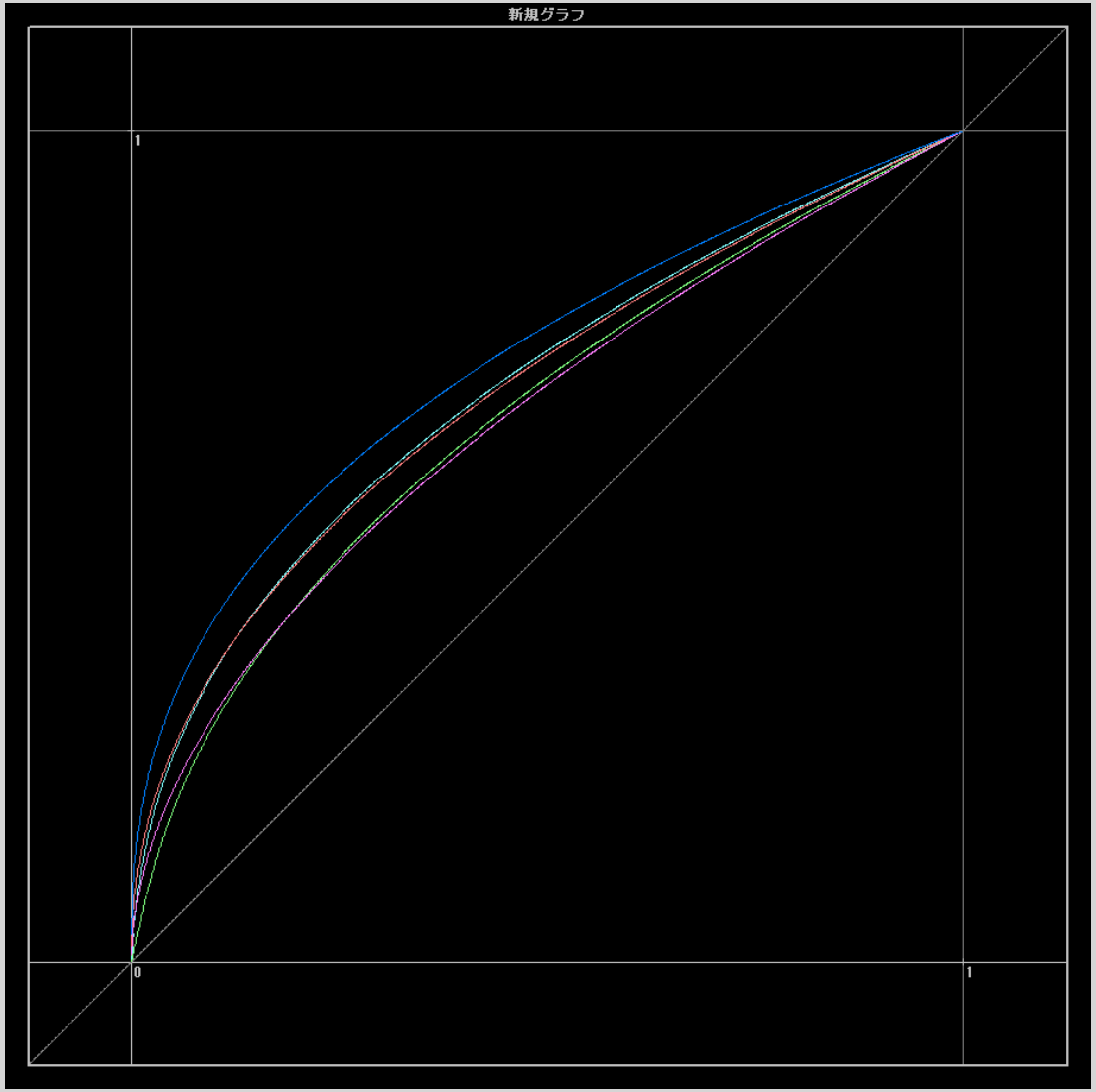
- DCI-P3
  - $V = L^{1/2.6}$




Silicon Studio ©Silicon Studio Corp., all rights reserved.

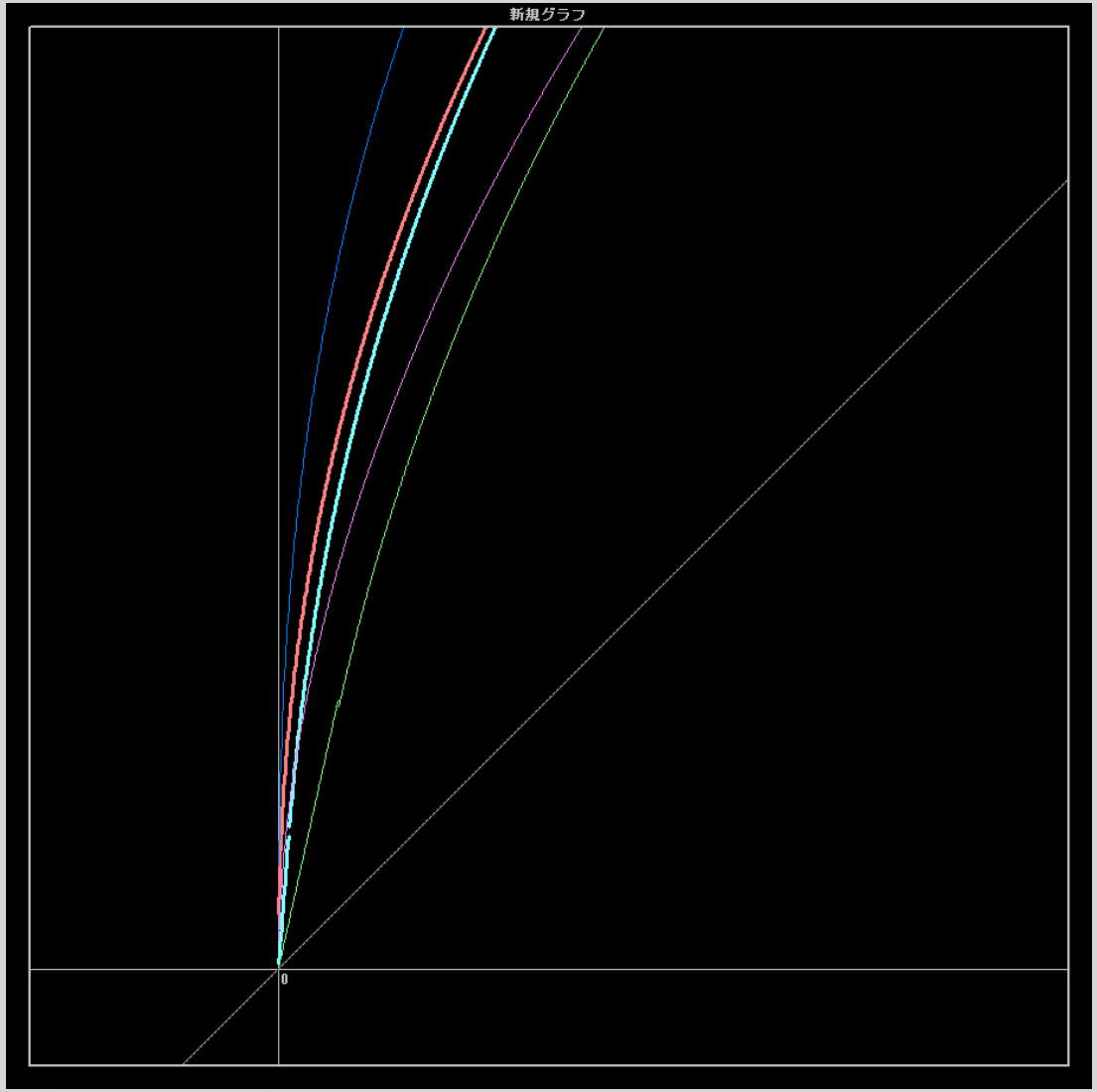
# 暗部の比較(BT.709, γ1.9, sRGB, γ2.2, DCI-P3)

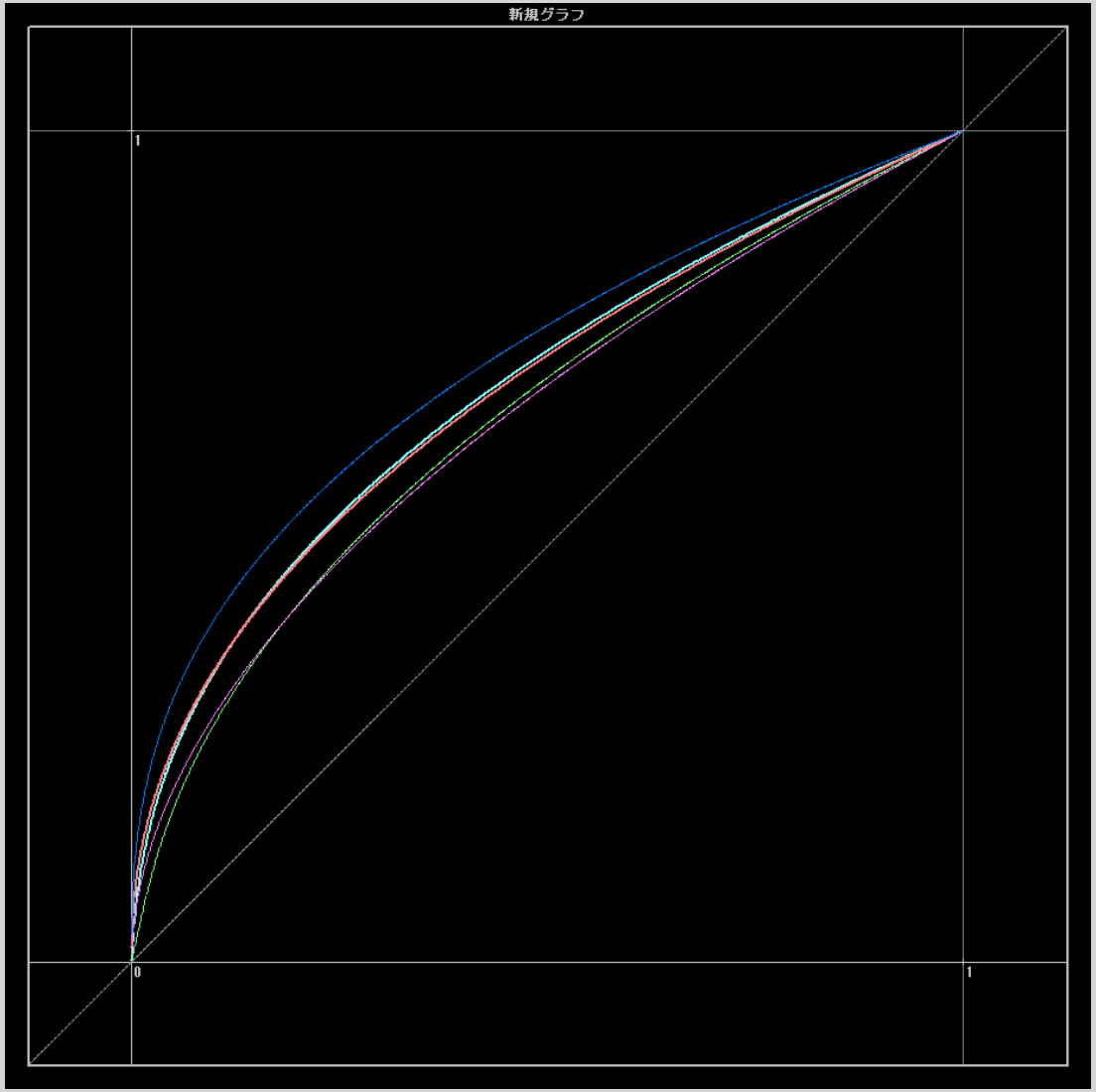



Silicon Studio ©Silicon Studio Corp., all rights reserved.

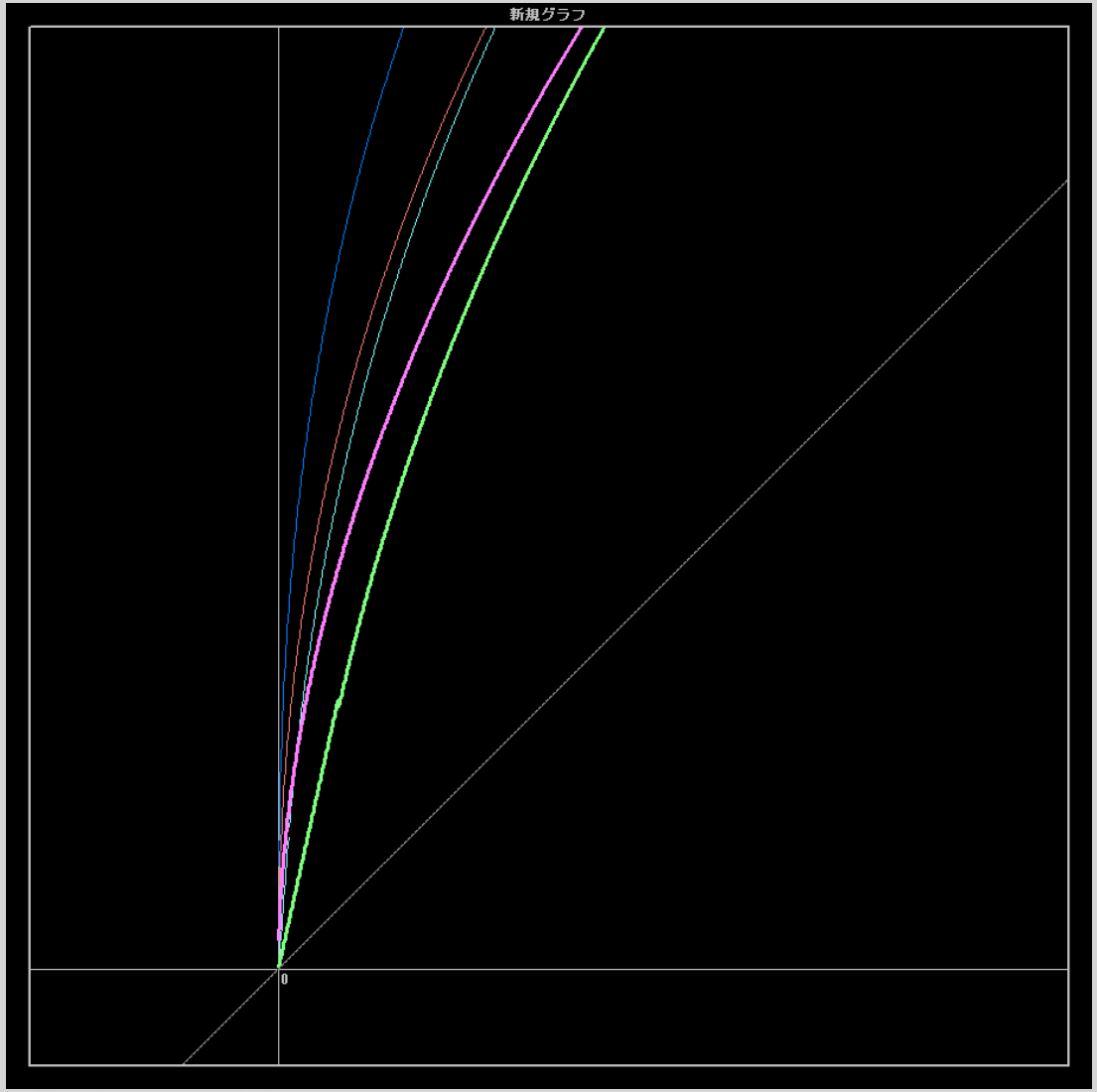
# 単一ガンマ近似との比較：sRGBとγ2.2補正

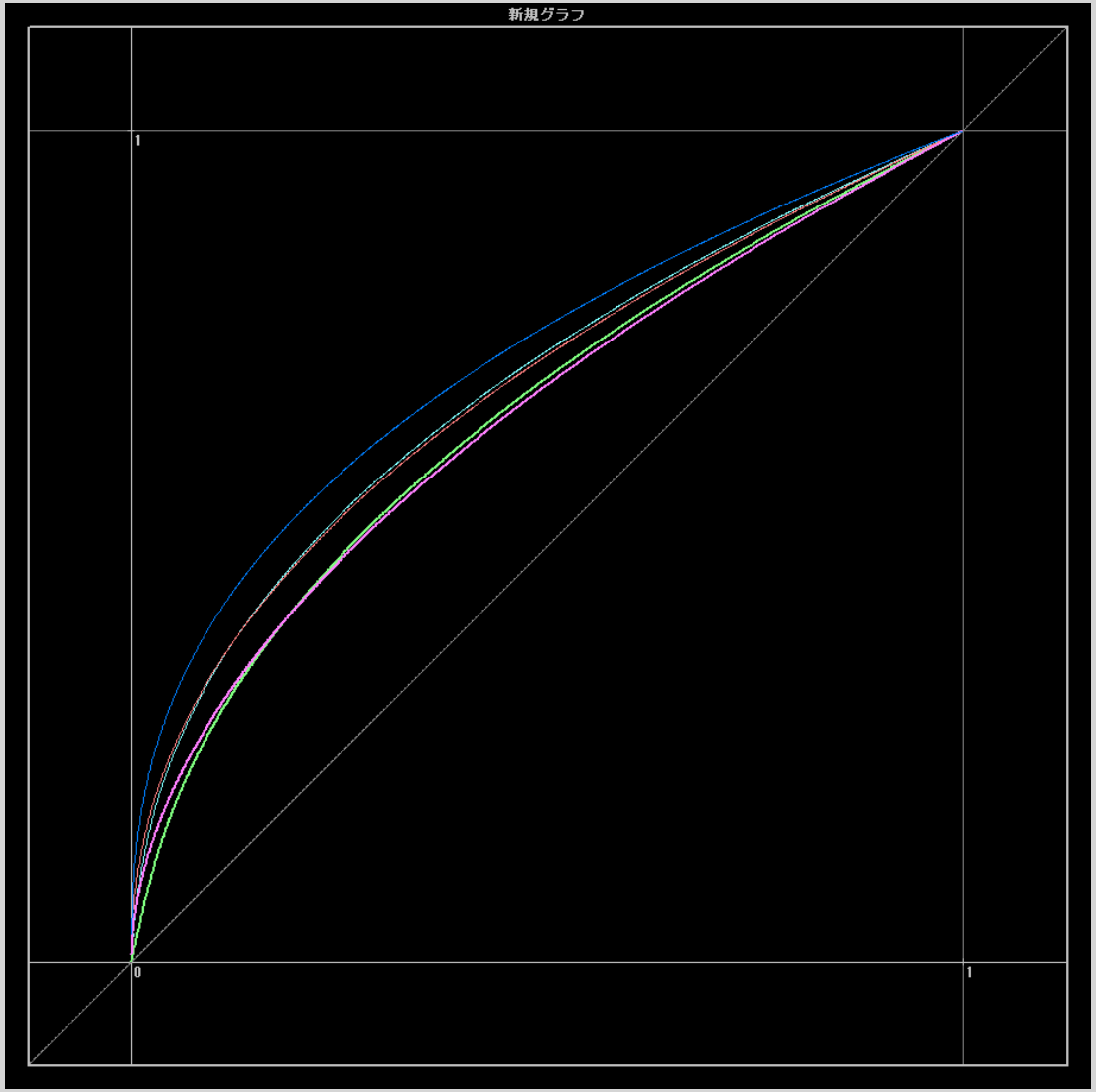



Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 単一ガンマ近似との比較：BT.2020とγ1.9補正




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 色空間を構成する要素

- 色域(Gamut)
- 伝達関数(Transfer Function)
- 輝度条件や鑑賞条件、ビット深度等


©Silicon Studio Corp., all rights reserved.

# 輝度条件や鑑賞条件、ビット深度等

- 規格によっては細かく規定されている
  - ただし実際の運用で必ずしも準拠しているとは限らない
- 例えばディスプレイ輝度は
  - sRGB: 80cd/$m^2$(=nt: nits)
  - Adobe RGB: 160cd/$m^2$
  - BT.709/BT.2020のテレビ出力(BT.1886): 100cd/$m^2$
  - DCI-P3: 48cd/$m^2$
- 他にも周囲環境の明るさや色温度など

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 本セッションで頻繁に参照する規格等

- BT.709/BT.2020
  - Recommendation ITU-R BT.709/2020あるいはRec. ITU-R BT.709/2020の略
    - ITU-R(ITU Radiocommunication Sector: 国際電気通信連合の無線通信部門)の勧告
    - Rec.709/Rec.2020とも略される
  - HDTV(BT.709)やUHDTV(BT.2020)のSDR信号規格
    - 伝達関数は暗部に線形領域をもつγ1.9程度のカーブ
      - カメラ側ガンマ補正が由来
    - ただしテレビ側の出力は一般的にBT.1886(Rec. ITU-R BT.1886)
  - 頻繁に(むしろ主に)色域の説明に使われる
    - その場合色域だけを指している事が多い
    - 本セッションでもBT.709やBT.2020は主に色域を指します

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 本セッションで頻繁に参照する規格等

- BT.2100
  - Recommendation ITU-R BT.2100
  - HDTV(2K)やUHDTV(4k/8K)のHDR信号規格
    - 色域はBT.2020と同じ
      - 色域だけを指す場合はBT.2020が参照されることが多い
    - 伝達関数はPQもしくはHLG(任意)
      - PQはEOTF側で、HLGはOETF側で定義されている
        - » 本セッションでは扱わない(シリコンスタジオによるHDR出力対応の理論と実践 [Kawase16])
      - PQとHLGの相互変換等も可能(一部の情報は失われる)
  - HDRの色空間定義を含むが一般にはあまり参照されない
    - 色空間としてはBT.2100 PQではなくBT.2020 PQと表記されることが多い
      - 色域はBT.2020という名称で参照されることが多いため

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 本セッションで頻繁に参照する規格等

- BT.1886
  - Recommendation ITU-R BT.1886
  - HDTV(BT.709)やUHDTV(BT.2020)のテレビ出力側の変換関数
    - EOTFおよびテレビ側の映像調整に関する勧告
    - 伝達関数(EOTF)は線形領域のないγ2.4
      - マスターモニタのデファクトスタンダードが由来
    - 現在の家庭用テレビのリファレンス
  - 本セッションでも主に家庭用テレビのEOTF(γ2.4)を指します

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 本セッションで頻繁に参照する規格等

- PQ(Perceptual Quantizer)
  - Dolbyが開発
  - 規格としてはSMPTE ST-2084あるいはBT.2100のPQ EOTF
    - SMPTE(Society of Motion Picture and Television Engineers: 米国映画テレビ技術者協会)の定める伝達関数
    - Recommendation ITU-R BT.2100の定める伝達関数の一つ
  - 最大10,000cd/$m^2$(nt: nits)の絶対輝度を表現
    - 現状他の全ての伝達関数は相対輝度
      - PQ以外は信号の最大値＝ディスプレイに表示できる最大輝度
  - PQ自体は色域を定義しない
    - 本セッションでもPQは伝達関数(EOTF/OETF)のみを指します

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 本セッションで頻繁に参照する規格等

- sRGB
  - MicrosoftとHewlett-Packardが開発
  - IEC(International Electrotechnical Commission: 国際電気標準会議)の定める標準色空間
  - BT.709と同じ色域をもつ
    - ただし伝達関数(ガンマ)はBT.709とは異なるので注意
      - sRGBは暗部に線形領域をもつγ2.2程度のカーブ
      - BT.709は暗部に線形領域をもつγ1.9程度のカーブ
  - 本セッションでは主にsRGBガンマを指します
    - 伝達関数を含む色空間としても使用
    - 色域だけを示す場合は主にBT.709を使用


©Silicon Studio Corp., all rights reserved.

# 本セッションで頻繁に参照する規格等

- scRGB
  - MicrosoftとHewlett-Packardが開発
  - IEC(International Electrotechnical Commission: 国際電気標準会議)の定めるsRGBの拡張色空間
  - sRGB(BT.709)と同じ原点／白色点をもつ
  - 負の数も含む線形色空間
    - 本来のRGB値の範囲は-0.5～7.4999と定められている
    - 負の数により内部データとしては広色域も格納できる
    - レンダリングでの運用上はBT.709色域のHDRフォーマットとみなせる
      - 従来の一般的な線形HDRレンダリングフォーマット
  - 他にも非線形等いくつか類似の色空間がある
  - 本セッションでは実数の線形sRGB色空間として扱います


©Silicon Studio Corp., all rights reserved.

# 前提として押さえておきたいこと

本セッションの目的
色空間
前提として押さえておきたいこと
基本的なレンダリングフローの検討
その他の考慮すべきこと
互換性重視レンダリングフローの構築
まとめ

Silicon Studio ©Silicon Studio Corp. all rights reserved!

# 前提として押さえておきたいこと

- トーンマップのHDR出力拡張
- 白レベル(Paper White Level)の調整
- SDR時のシステムガンマとの互換性
- 色域変換と色域マッピング
- ビデオ出力設定の仕様

- “シリコンスタジオによるHDR出力対応の理論と実践”
  CEDEC 2016 [Kawase16] も参照

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 前提として押さえておきたいこと

- トーンマップのHDR出力拡張
- 白レベル(Paper White Level)の調整
- SDR時のシステムガンマとの互換性
- 色域変換と色域マッピング
- ビデオ出力設定の仕様


©Silicon Studio Corp., all rights reserved.

# トーンマップのHDR出力拡張

- 従来のトーンマップはHDRデータをSDR領域に圧縮
  - 最大1.0に飽和
- HDRではこれを任意の幅に拡張する

- トーンマップのHDR拡張に際して重要なこと
  - SDRとの互換性を保つ
  - あらゆる最大輝度への対応
    - さまざまな最大輝度のHDRテレビに対応するため

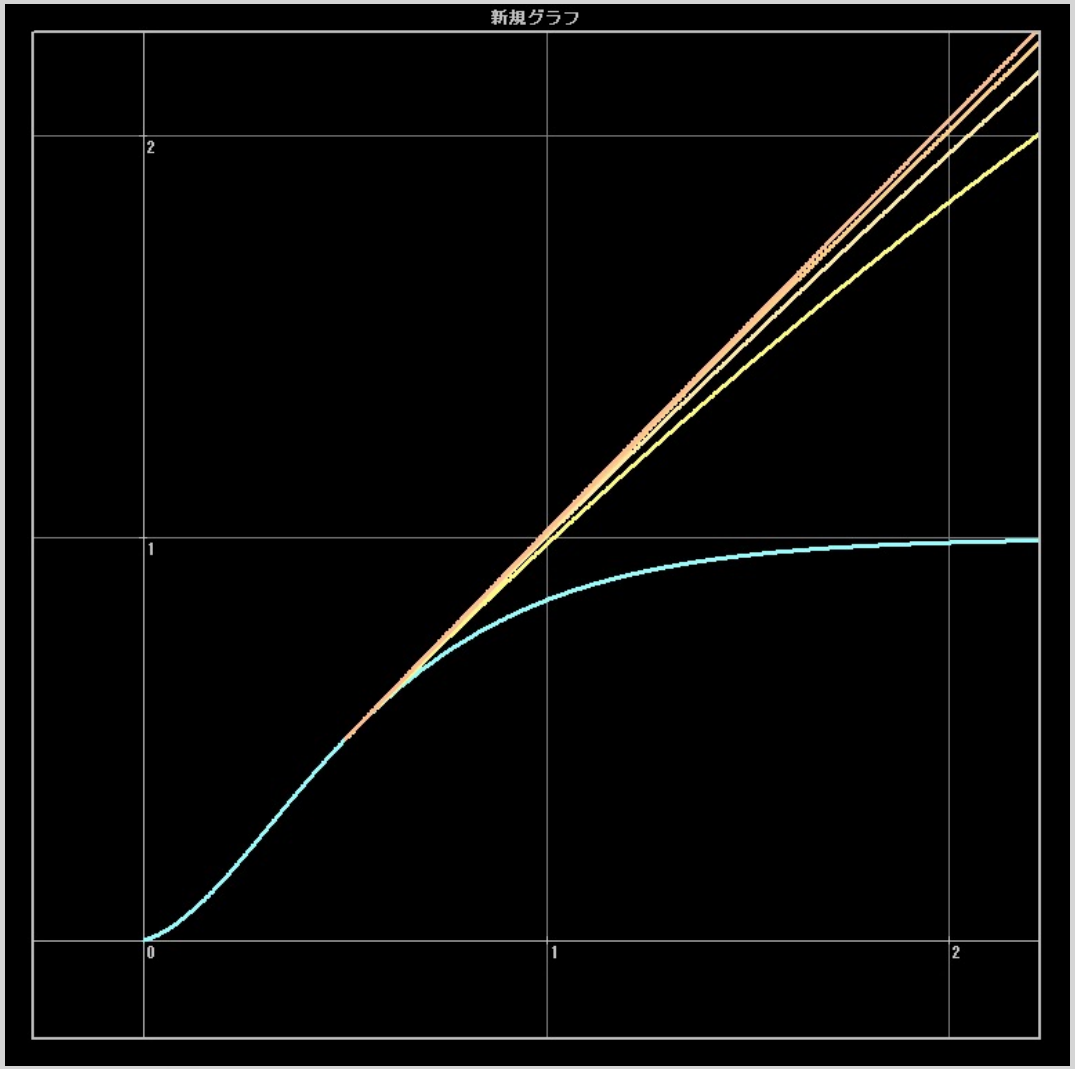

©Silicon Studio Corp., all rights reserved.

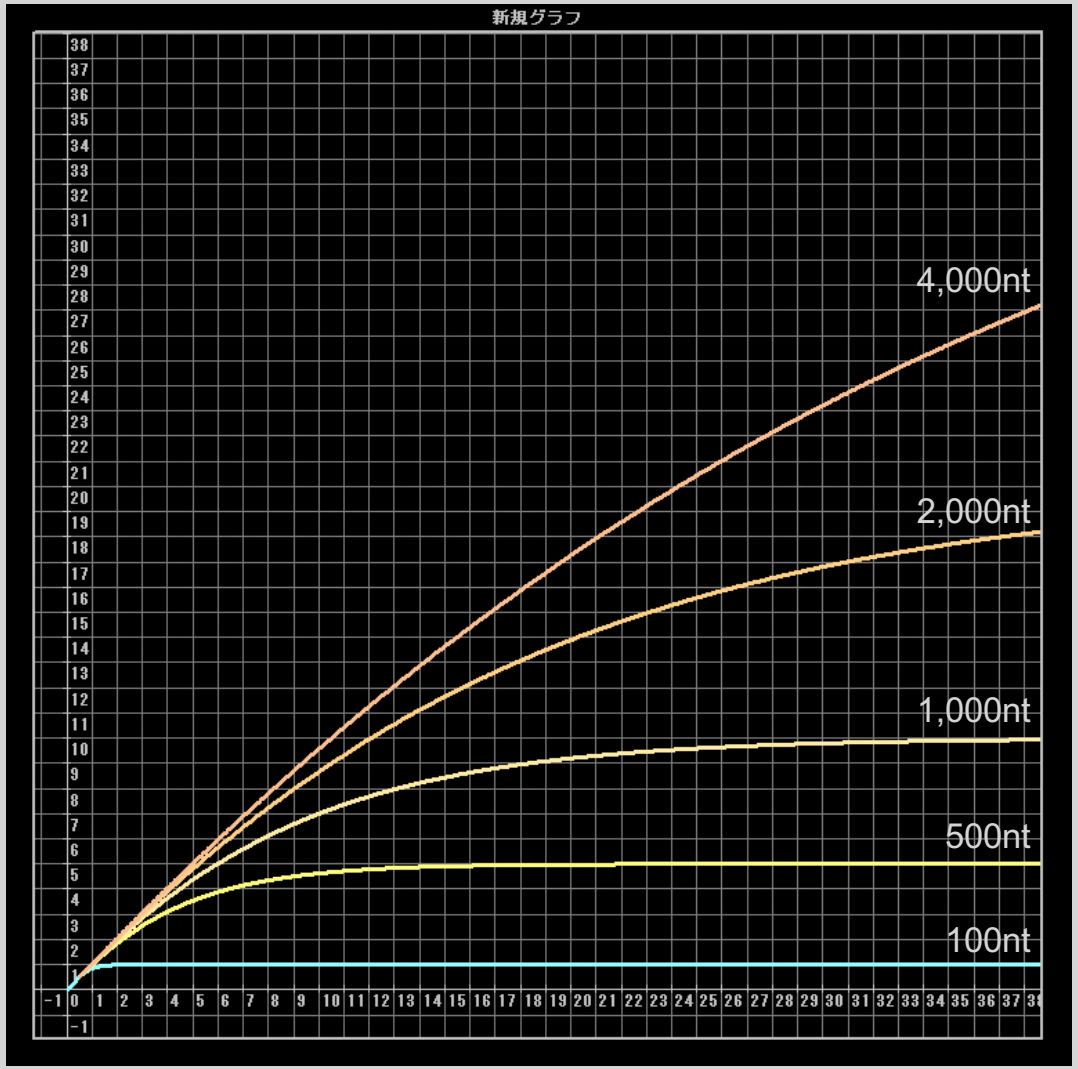
# トーンマップのHDR出力拡張

- HDR専用に開発された関数はSDRと互換性が低い
  - HDR専用に新たなパラメタの調整が必要
  - 原理的に同じような見た目に出来ない
- SDRとの互換性を損なわないトーンマップが必要
  - SDRトーンマップの輝度圧縮領域のみを拡張
    - イメージの支配的な構成部分である暗部領域を変えない


©Silicon Studio Corp., all rights reserved.

# HDR出力拡張したトーンマップ

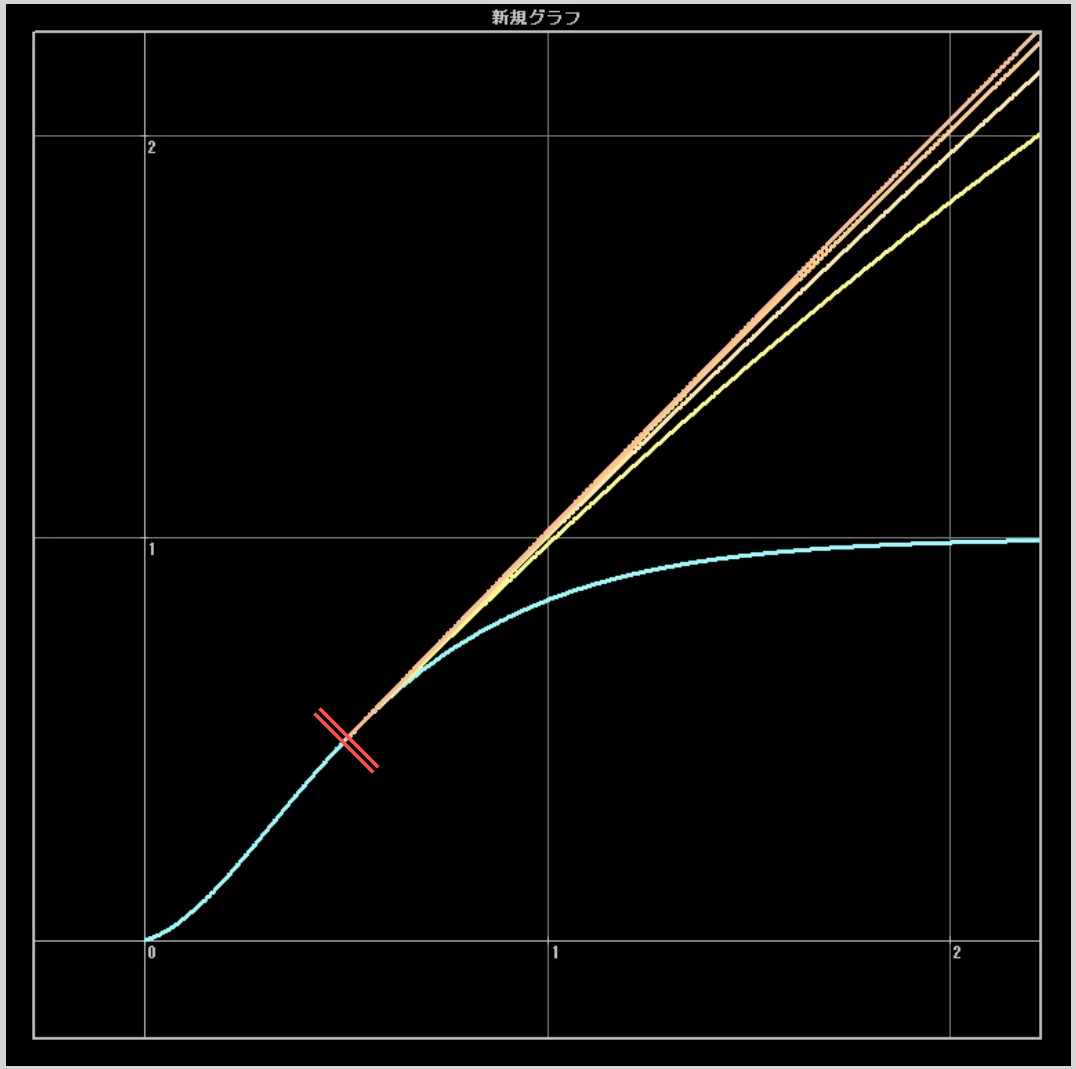



Silicon Studio ©Silicon Studio Corp., all rights reserved.

# HDR拡張トーンマップの作り方

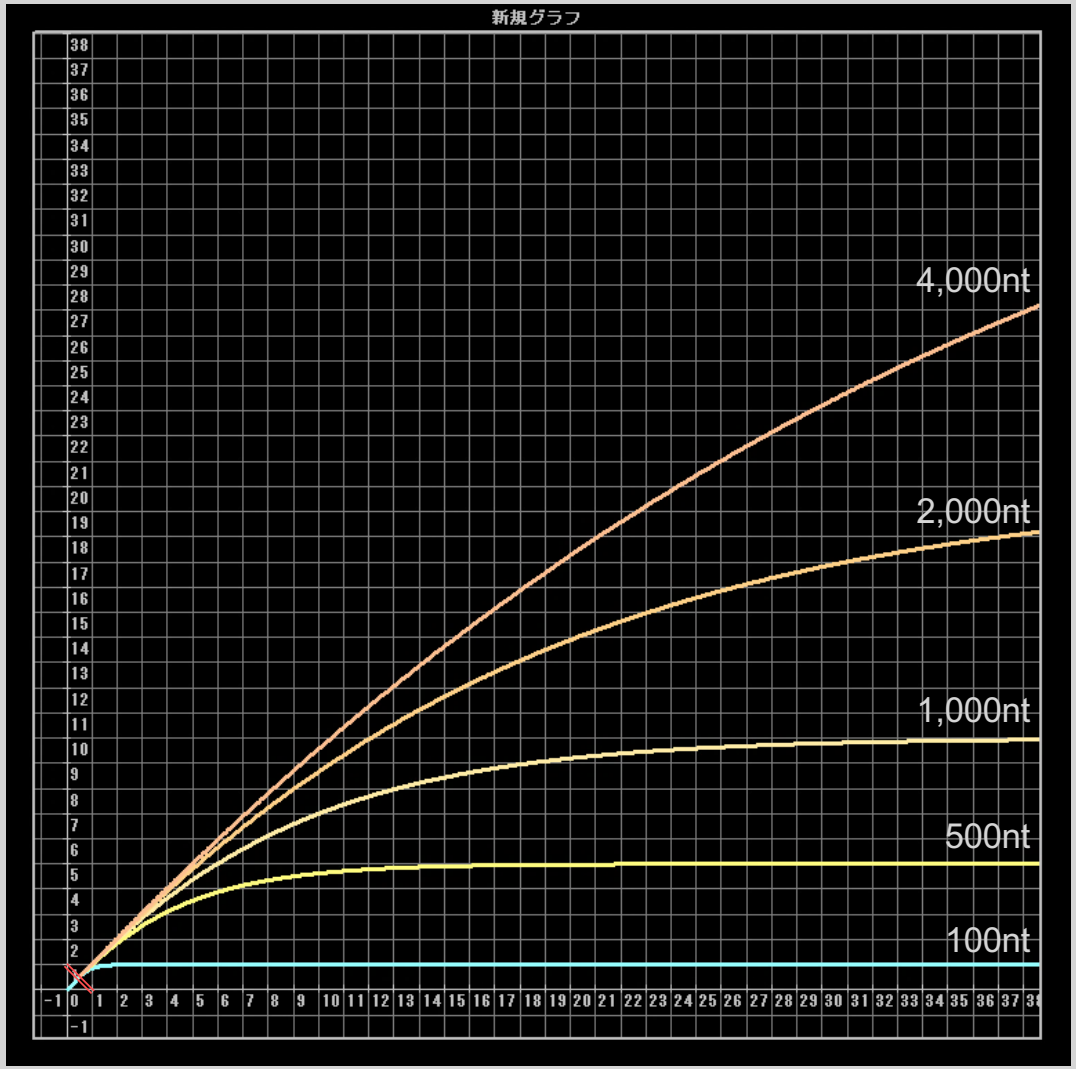
- 飽和部分から上下を分離
  - 最も膨らんでいる部分
- 暗部はSDRと共通
- 高輝度領域だけを拡大
  - 最大出力輝度に合わせて飽和




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# HDR拡張トーンマップの作り方

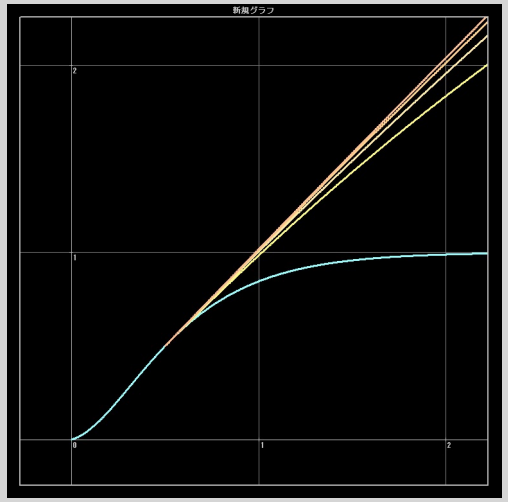
- 飽和部分から上下を分離
  - 最も膨らんでいる部分
- 暗部はSDRと共通
- 高輝度領域だけを拡大
  - 最大出力輝度に合わせて飽和

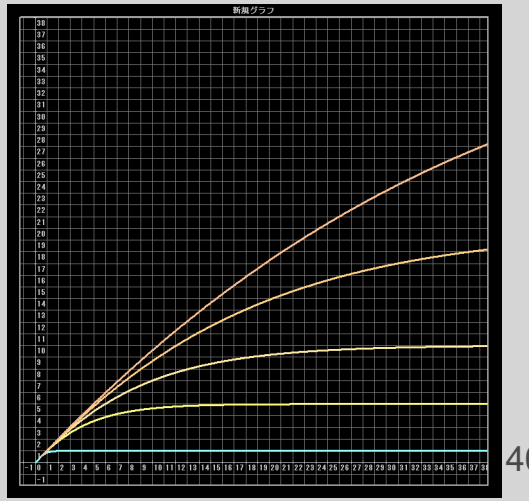



Silicon Studio ©Silicon Studio Corp., all rights reserved.

# HDR拡張トーンマップ

- 支配的な暗部領域が同じなので互換性が高い
  - 共通のトーンマップパラメタを利用できる
  - SDR調整だけでも使用できる
- ほとんどあらゆるトーンマップに適用可能
  - いわゆるSカーブならすべて拡張可能
  - Reinhardのような飽和部分のみももちろん可
    - このような場合は分離せずに全体を拡大するだけ






©Silicon Studio Corp., all rights reserved.

SDRシミュレーション
実際の1/4の輝度で出力
Silicon Studio
Mizuchi41

HDRシミュレーション
SDRの4倍のダイナミックレンジ
暗い部分の絵作りは変化しない
Silicon Studio
Mizuchi42

# 前提として押さえておきたいこと

- トーンマップのHDR出力拡張
- 白レベル(Paper White Level)の調整
- SDR時のシステムガンマとの互換性
- 色域変換と色域マッピング
- ビデオ出力設定の仕様


©Silicon Studio Corp., all rights reserved.

# 白レベル(Paper White Level)の調整

- 規格上の標準輝度
  - sRGB: 80nt
  - BT.1886: 100nt
- 実際の製品の明るさは
  - PCモニタ：おおよそ150～350nt程度
  - 家庭用テレビ：おおよそ200～450nt程度

Silicon Studio ©Silicon Studio Corp., all rights reserved.

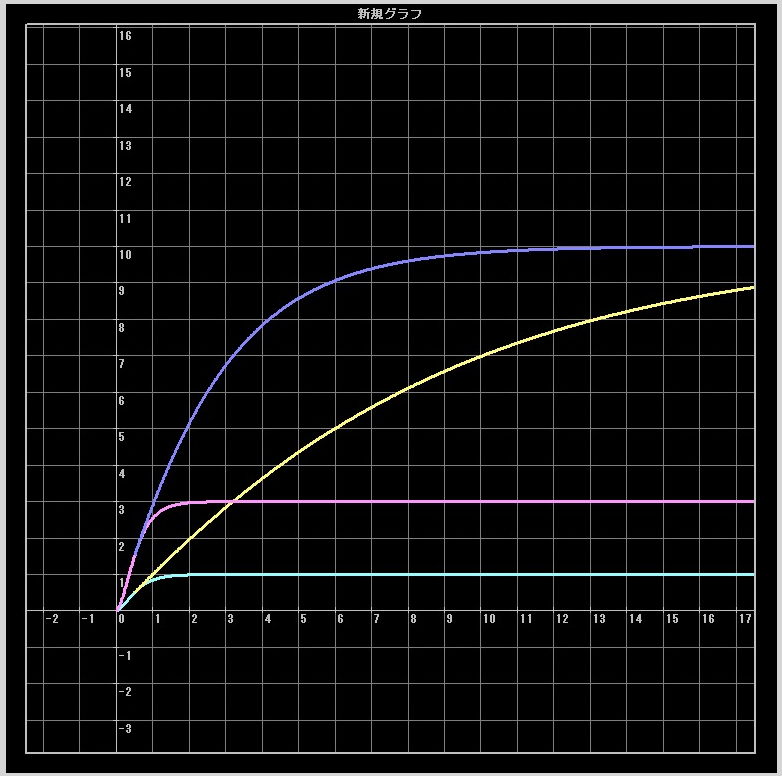
# 白レベル(Paper White Level)の調整

- 規格上の標準輝度は実情に即していない
  - 規格上のSDR輝度からHDR輝度を計算すると…
    - SDRと比較して1/2～1/4程度に暗くなる
- 実情に合わせた白レベルで出力するなら
  - 例えば80ntや100ntではなく300ntとみなす必要がある

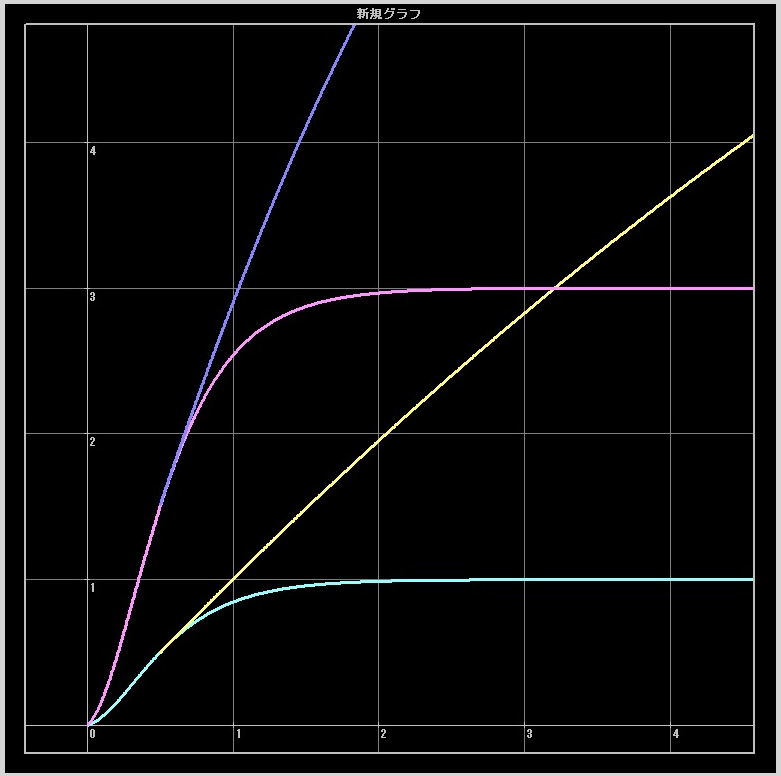

©Silicon Studio Corp., all rights reserved.

# 白レベル(Paper White Level)の調整

- 例えば以下のようにみなした場合
  - SDRテレビ輝度(白レベル)が300nt
  - HDRテレビ最大輝度が1,000nt




300nt基準のHDR出力
実情に合わせた調整




100nt基準のHDR出力
PQ(ST-2084)規格通り

300nt相当のSDR出力
SDRモニタの実情

100ntのSDR出力
BT.1886規格通り

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 白レベルの調整(トーンマップ)

- トーンマップ時の飽和輝度(相対輝度)
  - テレビの最大出力輝度は白レベルに関わらず固定
  - →白レベルに反比例して相対飽和輝度を下げる
    - $n = {}^{m}/_{w_c}$
      $n$ : 相対飽和輝度(1.0 = 白)
      $m$ : ディスプレイ最大絶対輝度(nt)
      $w_c$ : 白レベル(nt)
    - 白レベルが300ntで最大1,000-ntモニタなら
      - トーンマップ飽和は 1000 / 300 = ~3.333
    - 白レベル80ntで最大10,000ntなら125.0
      - sRGB規格の標準白レベルでPQの最大輝度に出力する場合

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 白レベルの調整(OETFエンコード)

- PQ OETFの入力は絶対輝度(nt: cd/m²)
  - 正確には1.0を10,000ntとみなす正規化値(nt値 / 10,000)
- PQ OETFのための絶対輝度を求める
  - 露出／トーンマップ済相対輝度(レンダリング結果)から絶対輝度(nt)へ換算
    - 露出／トーンマップ済レンダリング結果(相対輝度)は1.0が白なので
    - $L = w_c x$

      $L$ : 絶対輝度(nt)

      $x$ : 露出／トーンマップ済相対輝度(1.0 = 白)

      $w_c$ : 白レベル(nt)
- 換算後の絶対輝度$L$を正規化してPQ OETFを適用
  - $y = {}^{L}/_{10000} = {}^{w_c x}/_{10000}$
  - $V$ =PQ_OETF($y$)


©Silicon Studio Corp., all rights reserved.

# 白レベルの調整(OETFエンコード)

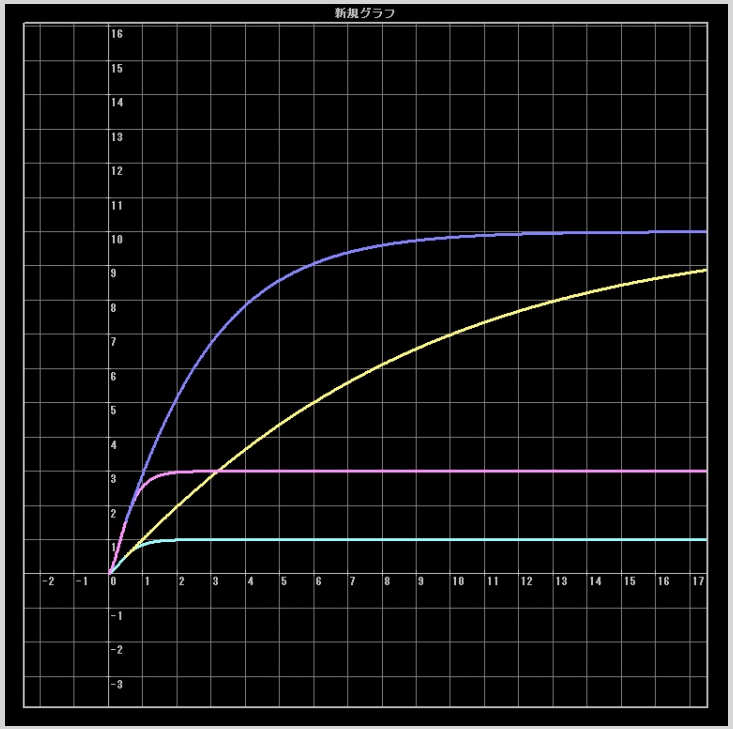
- 白レベルが300ntで最大1,000-nt HDRモニタの場合
  - トーンマップ飽和相対輝度は
    - $n = {}^{m}/_{w_c}$ = 1000 / 300 = ~3.333(前述)
  - トーンマップ飽和相対輝度をPQ OETF入力のための絶対輝度(nt)に換算
    - $L = w_c x$ = 300 x ~3.333 = 1000
    - 狙い通りモニタ最大輝度(1,000nt)に一致する

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 白レベル調整時の注意

- 出来れば200nt以下に抑える事を推奨
  - 高いほど白飛びが発生しやすい
  - せっかくのHDRの効果を活かせない

白レベルを300ntにした場合
事実上100ntの1/3で白飛びする




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# SDR出力のHDR化

- SDRとの差が埋まらない場合は逆にSDRを調整すべき
  - テレビがSDRとしては明る過ぎると考える
  - SDRでHDRシミュレーションを行う(MDR的な出力)
    - HDRにおける白レベル調整と逆の調整
    - HDR同様に1.0を超える拡張トーンマップを適用
    - OETF(sRGBガンマ補正)前に0.0～1.0に正規化
  - 既存タイトルではなく新規開発の場合は検討を強く推奨
- 可能ならユーザがカスタマイズできるように
  - 暗い部屋なら80nt～100ntベースでHDR感を存分に堪能したい！


©Silicon Studio Corp., all rights reserved.

# 白レベル調整／HDR化の注意

- 露出で調整しないように
  - 露出を変えるとSカーブの互換性を保持できない
    - 露出3倍→トーンマップ ≠ 1/3飽和トーンマップ→結果を3倍
  - →露出は常にSDR時の設定(18%グレー等)と同じに
    - あくまでトーンマップ飽和／OETF前スケールで調整する


©Silicon Studio Corp., all rights reserved.

SDR出力(SDRの100%出力が白)
従来のSDR
Silicon Studio
Mizuchi

HDRシミュレーション(最大400nt, 基準白レベル100nt)
実際の1/4(100nt/400nt)の輝度で出力
Silicon Studio
Mizuchi54

SDRシミュレーション(参考)
実際の1/4(100nt/400nt)の輝度で出力

# SDRによるHDRシミュレーションの注意点

- 8-bit RGBでは精度が不足する可能性が高い
  - 分解能の不足によるバンディング(シュブルール錯視)
  - ディザリングを行う
  - 可能なら10-bit RGBモード等を利用する


©Silicon Studio Corp., all rights reserved.

# 前提として押さえておきたいこと

- トーンマップのHDR出力拡張
- 白レベル(Paper White Level)の調整
- SDR時のシステムガンマとの互換性
- 色域変換と色域マッピング
- ビデオ出力設定の仕様


©Silicon Studio Corp., all rights reserved.

# SDRシステムガンマとの互換性

- 従来のSDRテレビにおけるOETFとEOTFの関係
  - コンテンツ側OETF
    - sRGB(一般的なGPUレンダリング)
      - 暗部線形領域をもつγ~2.2の補正
    - ビデオコンテンツではBT.709/BT.2020準拠の場合も
      - 暗部線形領域をもつγ~1.9の補正
  - 家庭用テレビ側EOTF
    - BT.1886(マスモニのデファクトスタンダードが由来)
      - 線形領域の無いγ2.4カーブ

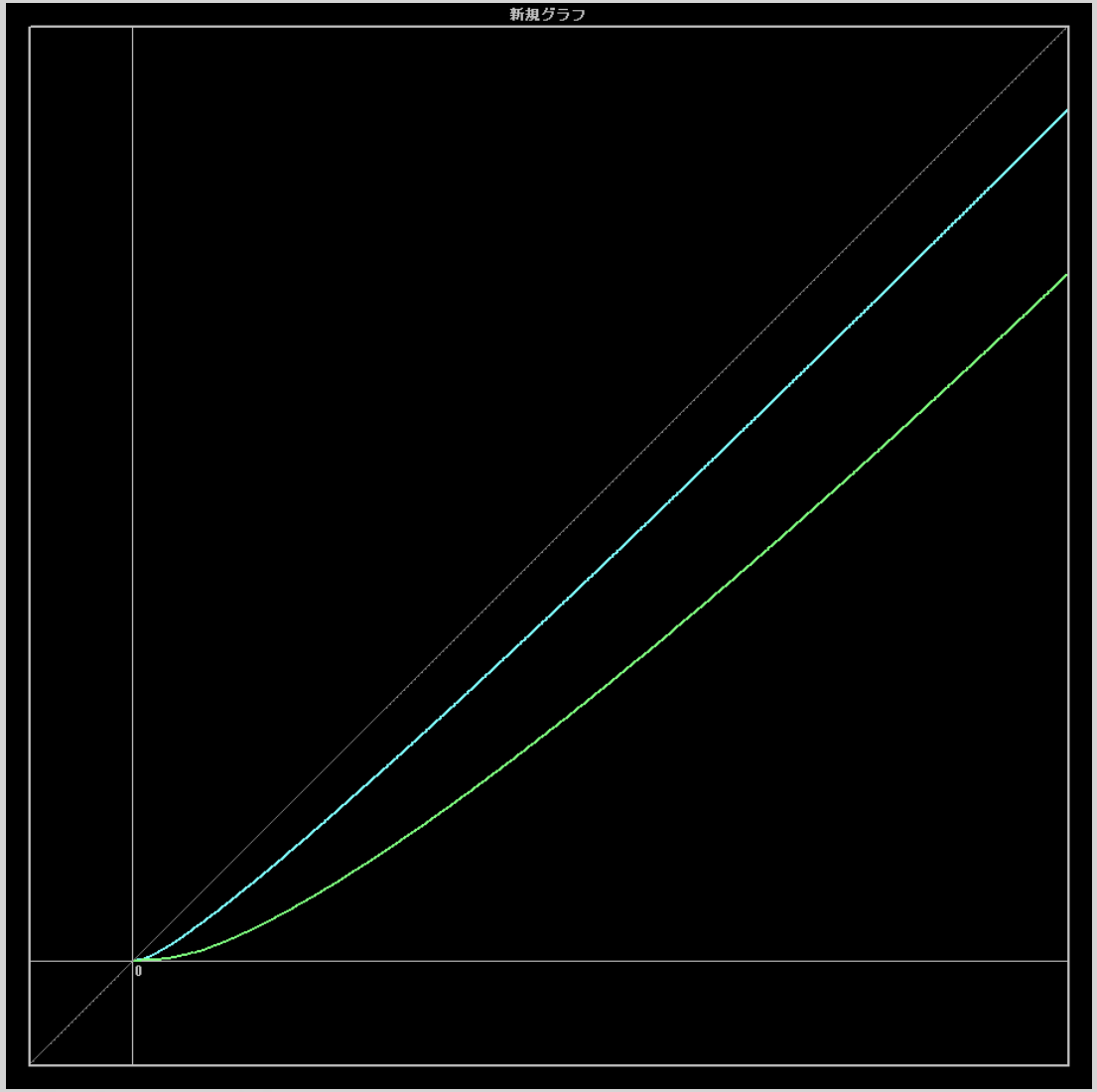

©Silicon Studio Corp., all rights reserved.

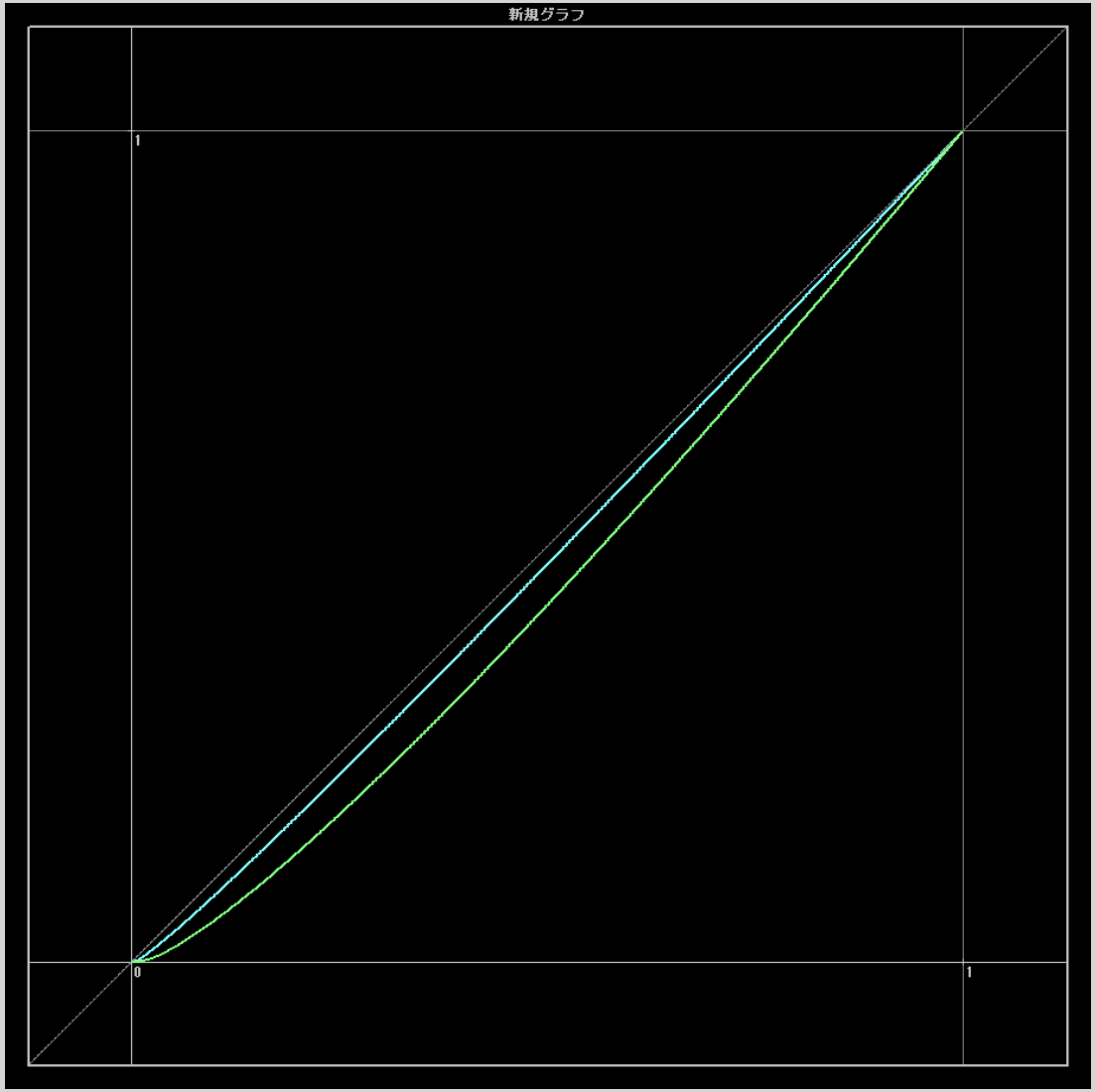
# SDRシステムガンマとの互換性

- OETFとEOTFが一致していない
  - →sRGBコンテンツではγ1.07程度の非線形関数
    - BT.709/2020コンテンツではγ1.25程度の非線形関数
  - システムガンマ(トータルガンマ)と呼ばれる
    - あるいはOOTF(Opto-Optical Transfer Function: 光－光伝達関数)

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# テレビOOTF: sRGB/BT.709 OETF→BT.1886 EOTF




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# SDRシステムガンマとの互換性

- SDR用OETFでは特に暗い部分の線形領域が影響
  - 最終出力では暗い領域のコントラストが高くなる
- HDRではこのような変換は行われない
  - OETFとEOTFが厳密に対応
  - →HDR出力対応の際、SDRと比較して暗い領域のコントラストが低下
    - 色が薄くなり灰色に浮いたように感じられる
  - ※厳密にはHDRでは別途OOTFが定義されている
    - EOTFベースのPQではモニタ側EOTFには影響なし
    - OETFベースのHLGでは別途システムガンマを補正する必要あり
    - 本セッションでは扱わない
      - “シリコンスタジオによるHDR出力対応の理論と実践” CEDEC 2016 [Kawase16]

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# SDRとの互換性を保つには

- SDR時のOOTFをシミュレーション
  - HDR拡張トーンマップ後PQ OETF前に
    - sRGB OETF→BT.1886(γ2.4)EOTF適用
    - 原則として1.0未満のSDR領域のみに適用
  - あるいはHDR拡張トーンマップ前に
    - 一時的にSDRトーンマップ空間で互換カーブを適用
      - SDR LUTを適用する場合に効率的(後述)
    - SDRトーンマップ(SDRトーンマップ空間)
    - sRGB OETF→BT.1886(γ2.4)EOTF適用
    - SDR逆トーンマップで線形HDRに戻す
    - HDR拡張トーンマップを適用

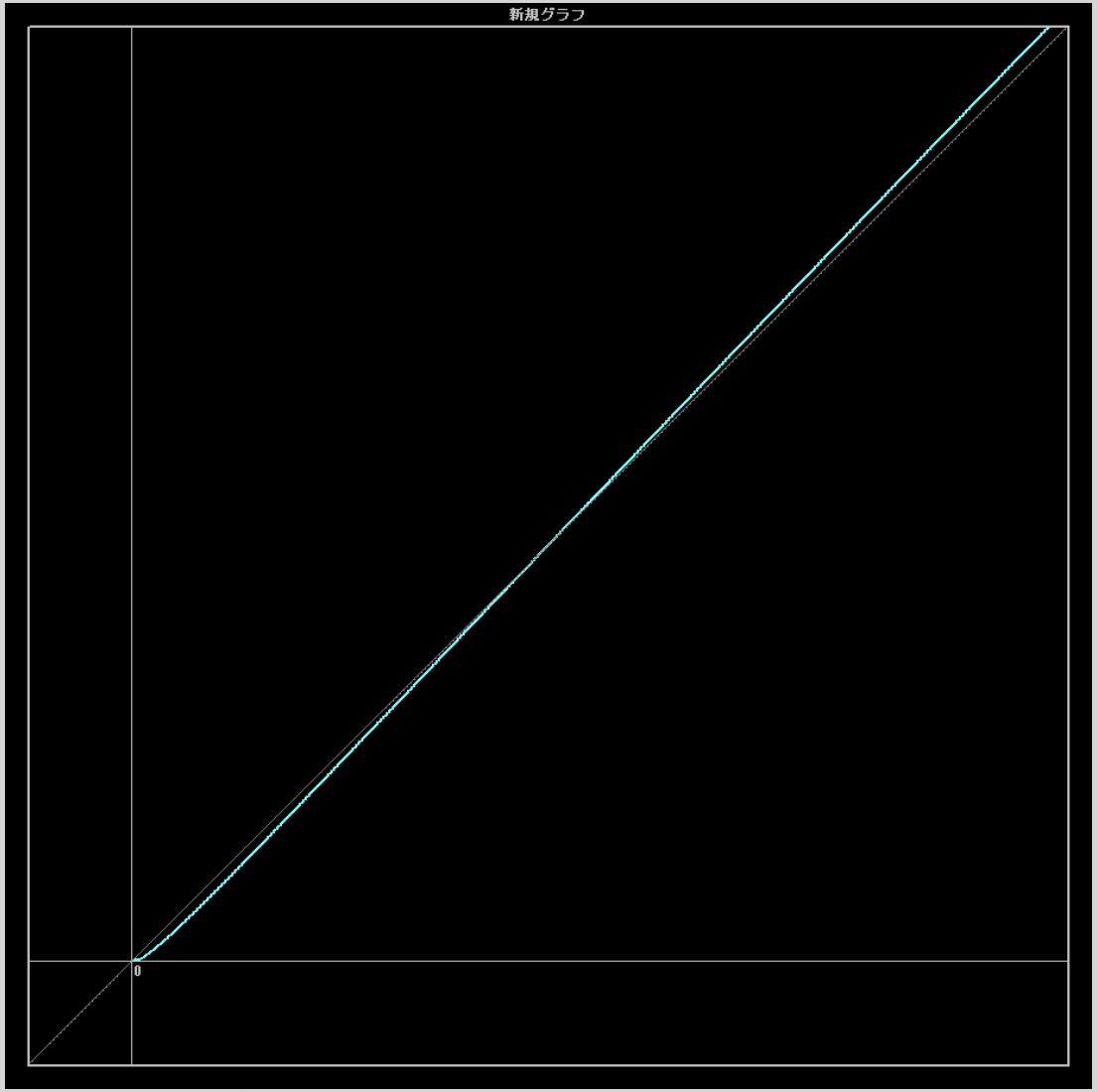
Silicon Studio ©Silicon Studio Corp., all rights reserved.

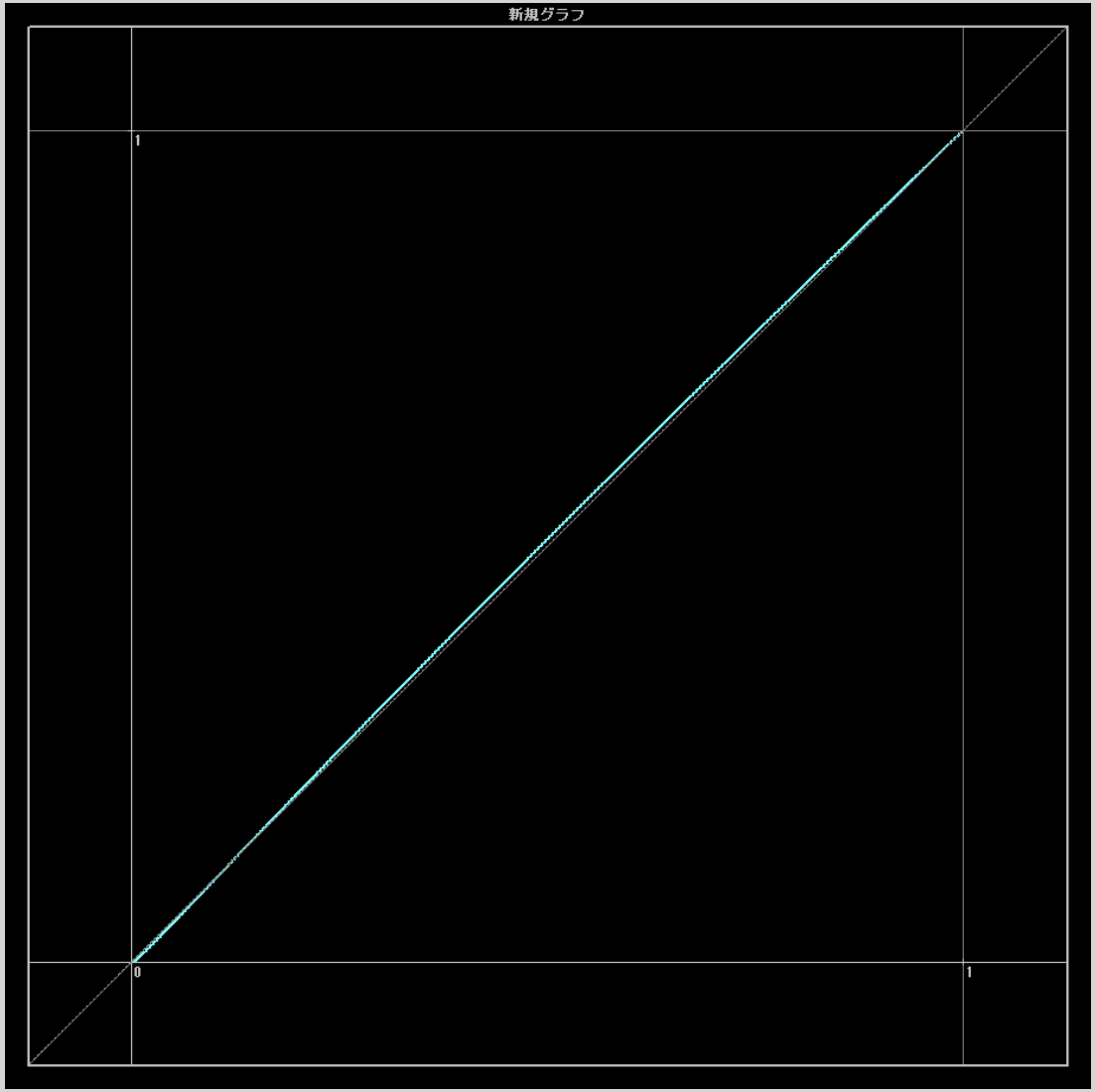
# PCモニタのSDRとの互換性は？

- いわゆるsRGBモニタの場合
  - ガンマ(EOTF)はBT.1886ではない
  - 線形領域の無いおおよそγ2.2のカーブ
    - プロユースの製品ではsRGBを厳密に出力できる場合も
      - その場合OOTFは線形になる
  - →一般ユーザ向け製品では線形部分の無いγ2.2とみなせる
    - この場合のSDR互換OOTFは
      - sRGB OETF→γ2.2 EOTF
      - 家庭用テレビのsRGB OETF→BT.1886(γ2.4)EOTFとは異なる

Silicon Studio ©Silicon Studio Corp., all rights reserved.

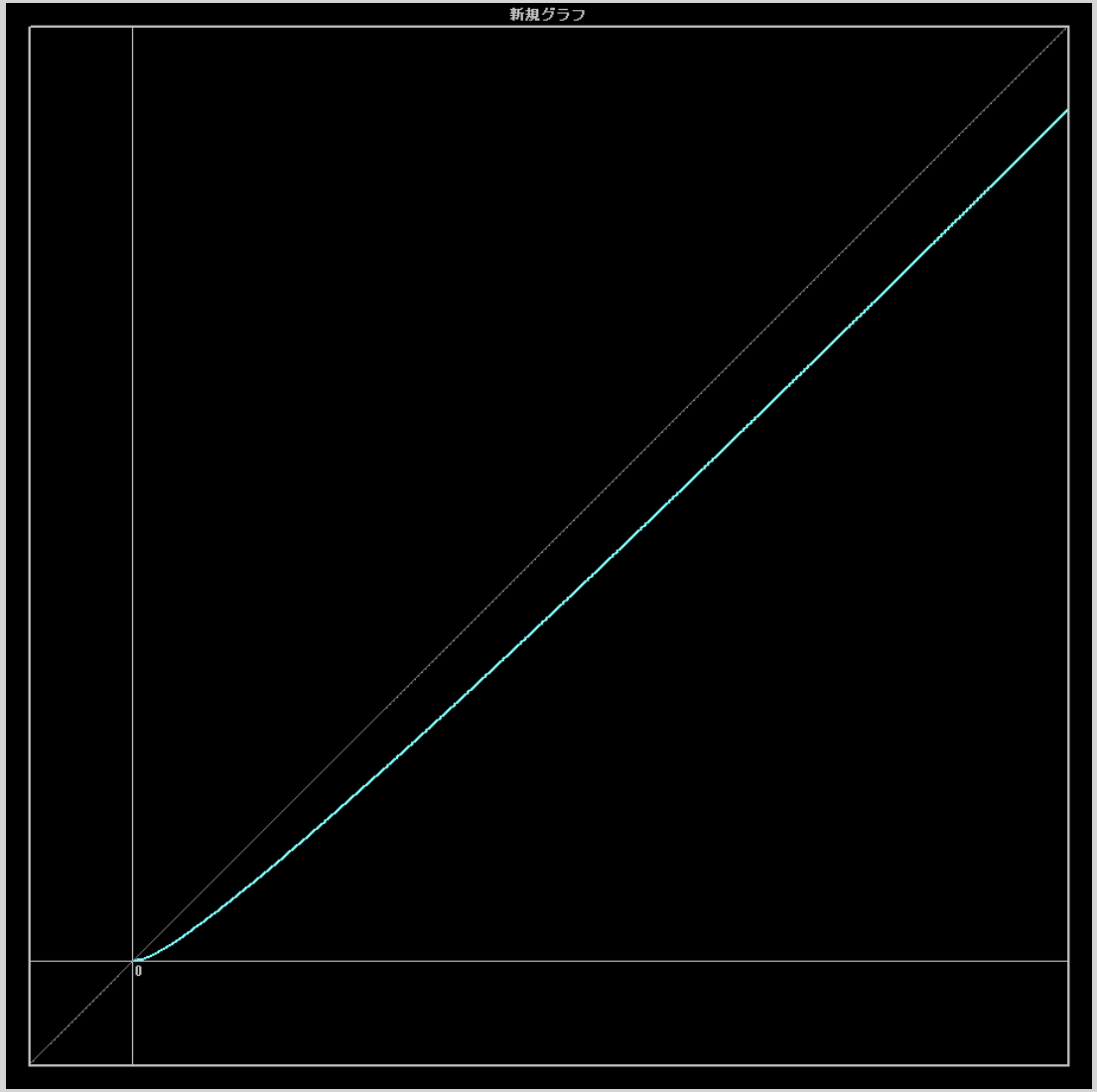
# PCモニタOOTF: sRGB OETF→γ2.2 EOTF

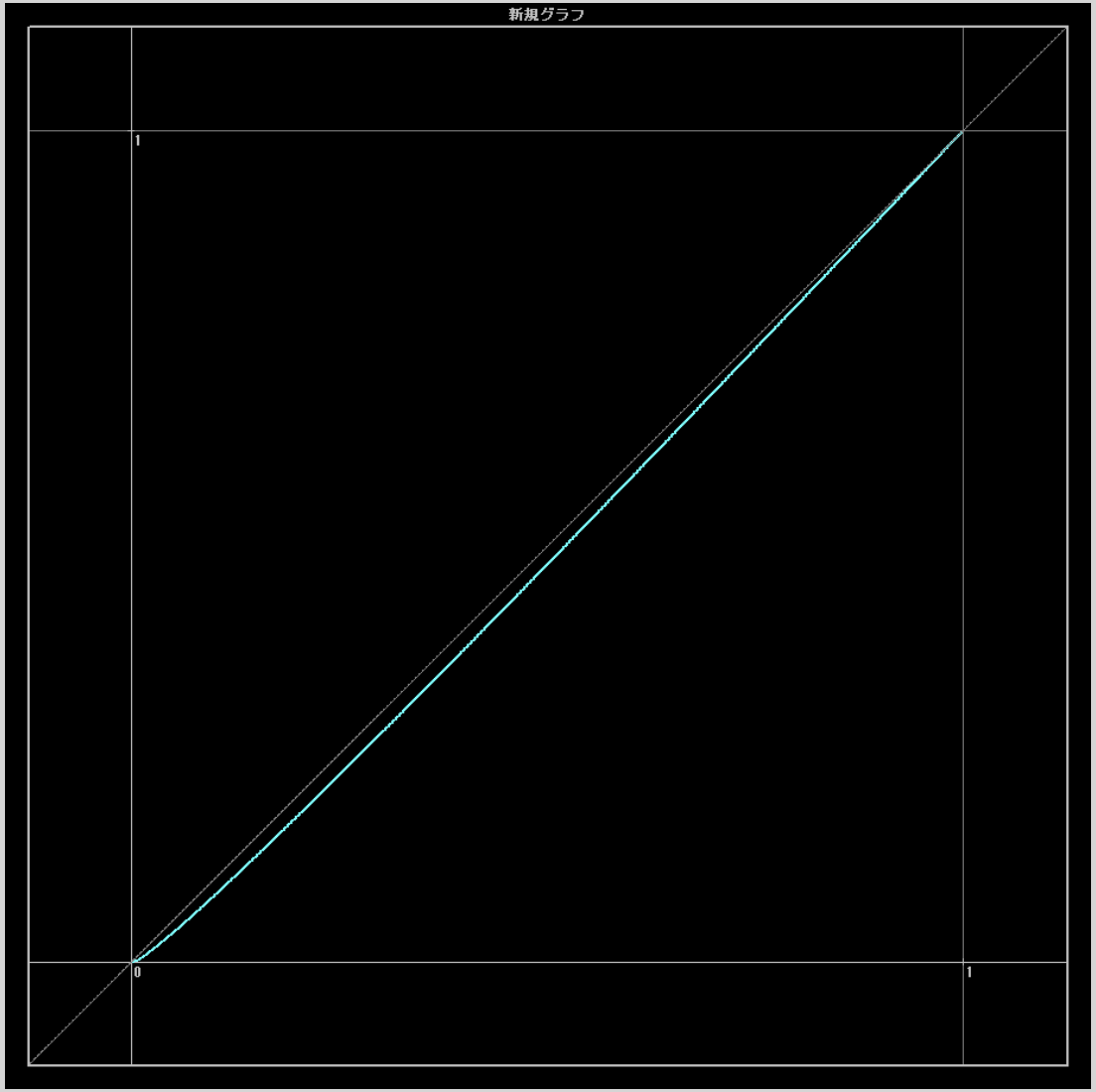



Silicon Studio ©Silicon Studio Corp., all rights reserved.

# テレビOOTF: sRGB OETF→BT.1886 EOTF




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OOTF有効／無効画像比較

- sRGB(SDR)モニタでは正しく表示できないので注意
  - SDR表示の時点で既にOOTFが適用されている状態


©Silicon Studio Corp., all rights reserved.

掲載の画像は権利上の理由などにより現在はご覧いただけません

OOTF無効

掲載の画像は権利上の理由などにより現在はご覧いただけません

家庭用テレビ向けOOTFシミュレーション
sRGB OETF→BT.1886(γ2.4)EOTF
SDR領域だけに適用した例

掲載の画像は権利上の理由などにより現在はご覧いただけません

OOTF無効

掲載の画像は権利上の理由などにより現在はご覧いただけません

PCモニタ向けOOTFシミュレーション

sRGB OETF→γ2.2 EOTF
SDR領域だけに適用

# システムガンマだけで互換性を保てるか？

- 家庭用SDRテレビはより派手な出力
  - 全体的にコントラストが高い
  - 全体的に彩度が高い
  - 全体的に色温度が高い
- SDRに限った話ではないがSDRの方が遥かに極端
  - 擬似的にHDRらしく拡張する製品が多い


©Silicon Studio Corp., all rights reserved.

# SDRテレビの実態との互換性

- テレビ／メーカーによって異なる
- 映像モードによっても異なる
  - ゲームモードはある程度派手なものが多い
  - PCモニタモード／グラフィックスモードは比較的忠実
- →汎用的な設定は難しい
  - 全体的なガンマ調整
  - 全体的な彩度調整
  - etc.

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 前提として押さえておきたいこと

- トーンマップのHDR出力拡張
- 白レベル(Paper White Level)の調整
- SDR時のシステムガンマとの互換性
- 色域変換と色域マッピング
- ビデオ出力設定の仕様

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 色域変換

- ある色域から別の色域への変換
  - ここではレンダリング色域から出力色域への変換
  - 基本は線形変換
    - 同じ色を別の色域で表現するための変換
    - 出力先で色域外となる色はクリップされる


©Silicon Studio Corp., all rights reserved.

# 色域変換行列の例

- BT.709からBT.2020への変換

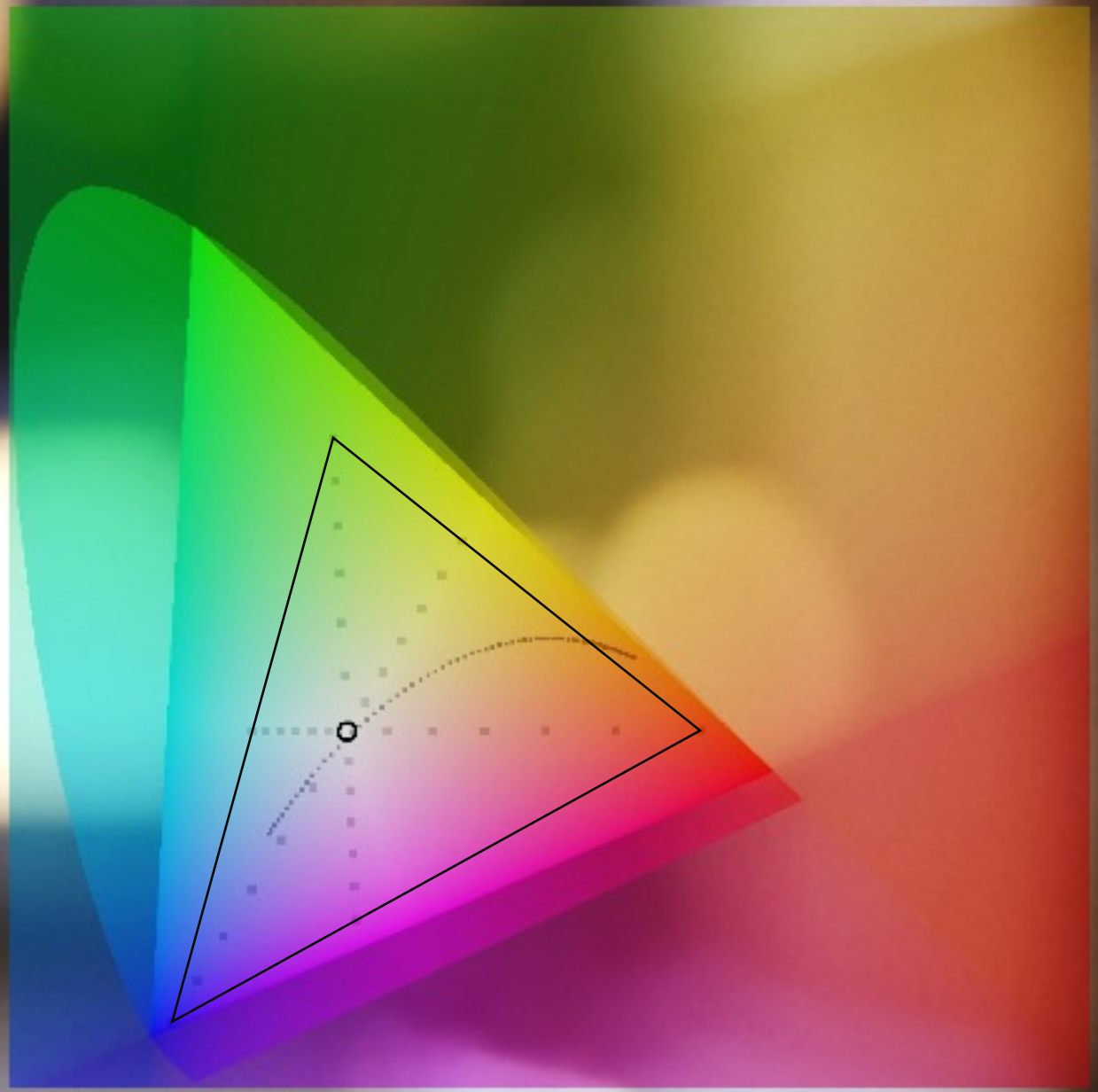
$$\begin{pmatrix} R_{2020} \\ G_{2020} \\ B_{2020} \end{pmatrix} = \begin{pmatrix} 0.6274039 & 0.3292830 & 0.0433131 \\ 0.0690973 & 0.9195406 & 0.0113623 \\ 0.0163914 & 0.0880133 & 0.8955953 \end{pmatrix} \begin{pmatrix} R_{709} \\ G_{709} \\ B_{709} \end{pmatrix}$$

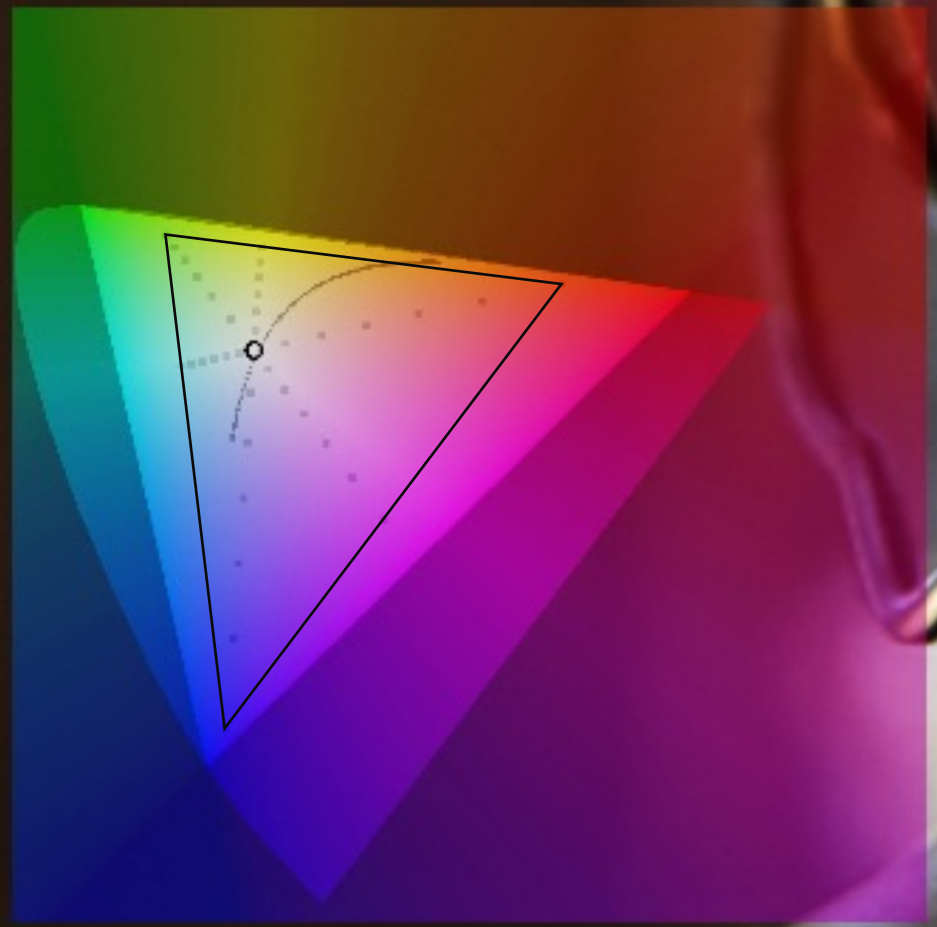
- BT.2020からBT.709への変換

$$\begin{pmatrix} R_{709} \\ G_{709} \\ B_{709} \end{pmatrix} = \begin{pmatrix} 1.6604912 & -0.5876410 & -0.0728499 \\ -0.1245506 & 1.1328998 & -0.0083494 \\ -0.0181508 & -0.1005789 & 1.1187296 \end{pmatrix} \begin{pmatrix} R_{2020} \\ G_{2020} \\ B_{2020} \end{pmatrix}$$


©Silicon Studio Corp., all rights reserved.

BT.709からBT.2020に線形変換
BT.709領域のみ使用



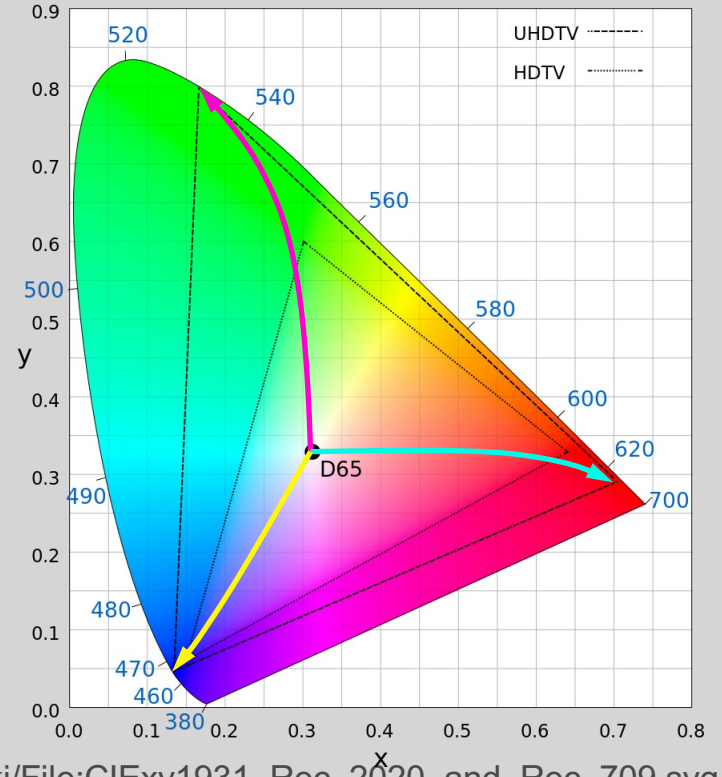


BT.2020からBT.709に線形変換
BT.709領域外はクリップ

掲載の画像は権利上の理由などにより現在はご覧いただけません

# 色域マッピング

- ある色域から別の色域への変換
  - 変換先で自然に見えるような最適化
    - 目的によって手法は異なる
    - 線形とは限らない
  - 物理的な整合性は失われやすい
  - 色相のズレが発生することもある




https://commons.wikimedia.org/wiki/File:CIExy1931_Rec_2020_and_Rec_709.svg

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 簡易的な色域マッピング

- 比較的簡単かつ無難な手法
  - 線形変換で出力先の色を計算
  - 変換前の彩度が低ければ変換先の色を採用
  - 変換前の彩度が高いほど変換前の色を採用
  - 彩度が高いほど急激なカーブで適用する
    - 彩度がある程度低い場合はほぼ線形変換の色にする
      - 全体の絵作りの印象を変えないため
    - 例えば彩度(0～1)を5乗程度にべき乗した値で補間
  - 簡単なパラメタで特性を制御可能


©Silicon Studio Corp., all rights reserved.

# 簡易的な色域マッピングの計算例

- BT.709で$RGB_{709}$=(0, 1, 0)のような純色(緑)
  - BT.2020に線形変換すると
    - $RGB_{2020}$=(0.3293, 0.9195, 0.0880)
  - 変換先と変換元RGBを変換元彩度の関数で補間
    - $RGB_{2020}$(0.3293, 0.9195, 0.0880)と$RGB_{709}$(0,1,0)を補間
  - 変換前の彩度は1.0(最大)のため
    - →RGB=(0,1,0)となり、線形変換からは大きく変化した緑の純色に
- BT.709で$RGB_{709}$=(1.0, 0.5, 0.5)のようなパステルカラー(ピンク)
  - BT.2020に線形変換すると
    - $RGB_{2020}$=(0.8137, 0.5346, 0.5082)
  - 変換前の彩度0.5の5乗($0.5^5$ = 0.03125)で変換元の色と補間すると
    - →RGB=(0.8195, 0.5335, 0.5079)となり、ほぼ線形変換と同じ(僅かに高彩度の)ピンクに


©Silicon Studio Corp., all rights reserved.

# 色域拡張マッピングの例

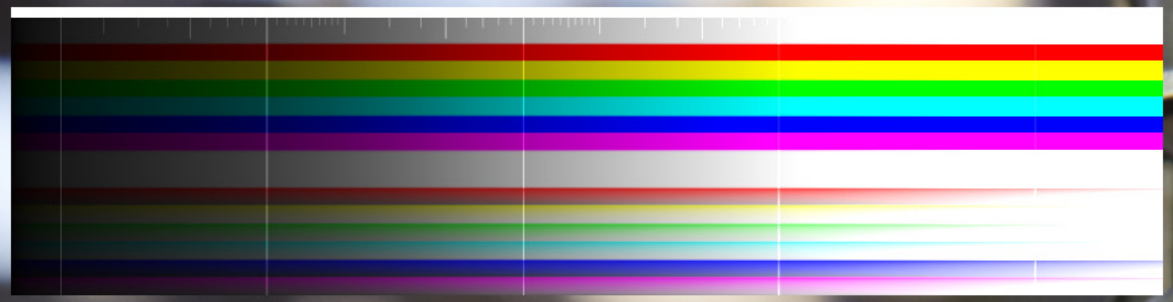
- BT.709でレンダリング
- BT.2020に出力
  - sRGBでは正しく確認できないので注意
- 色域マッピング無効／有効比較

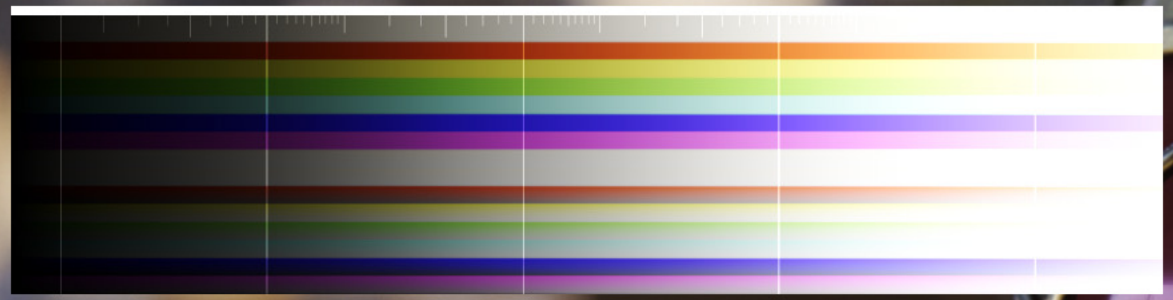
Silicon Studio ©Silicon Studio Corp., all rights reserved.

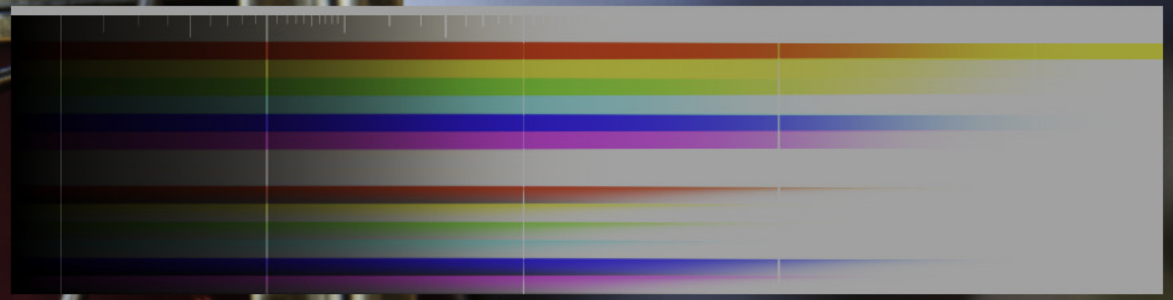
色域マッピング無効
BT.709領域のみ使用
Silicon Studio
Mizuchi82

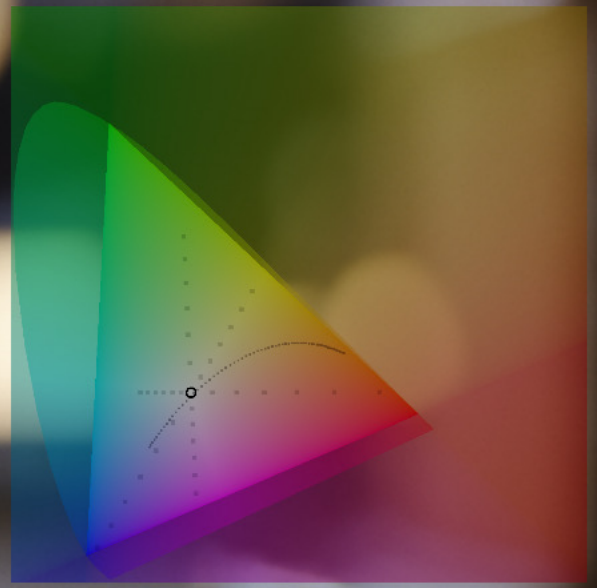
色域マッピング有効
BT.709からBT.2020へ拡張
Silicon Studio
Mizuchi

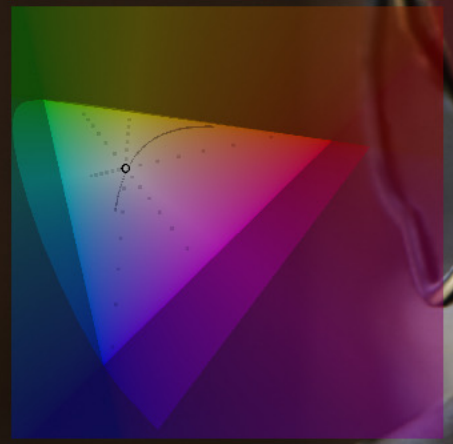
BT.709領域のみ使用

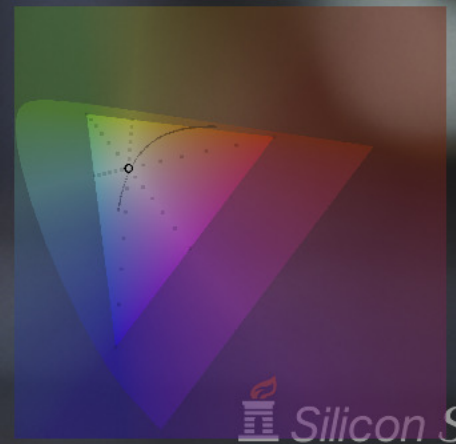






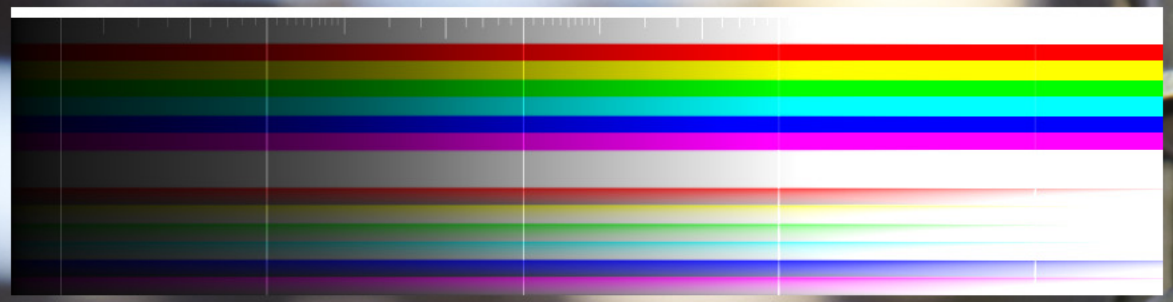


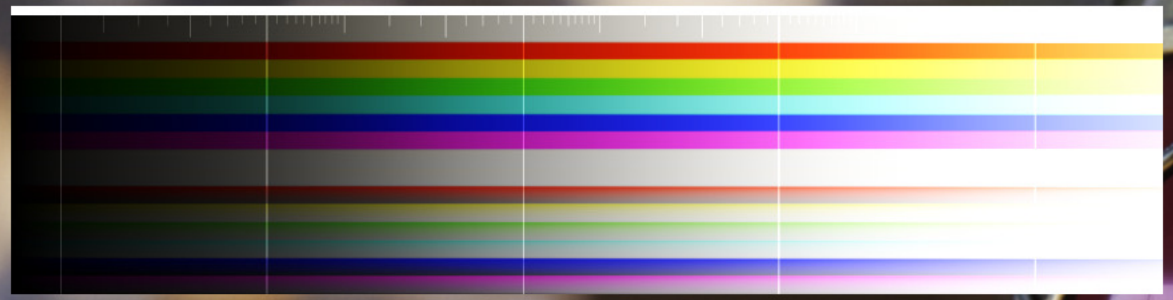


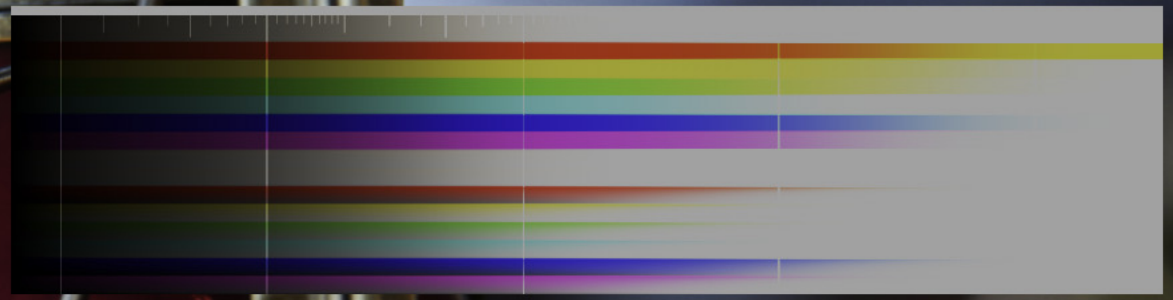


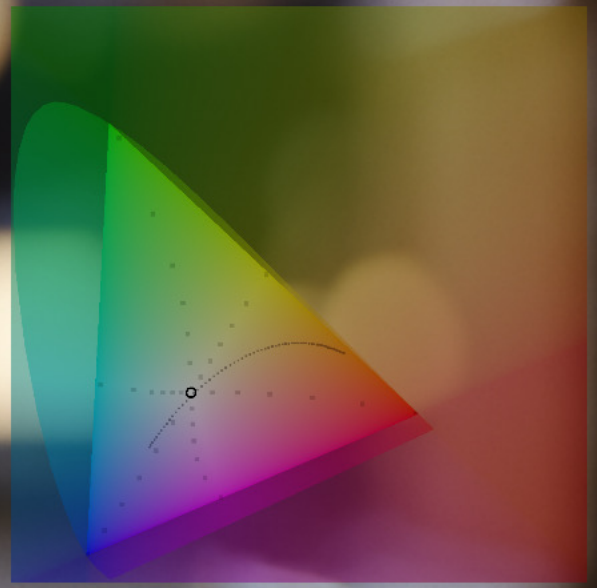
Silicon Studio
Mizuchi84

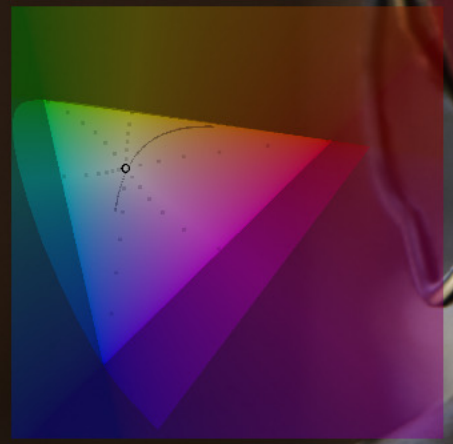
# BT.709からBT.2020へ拡張

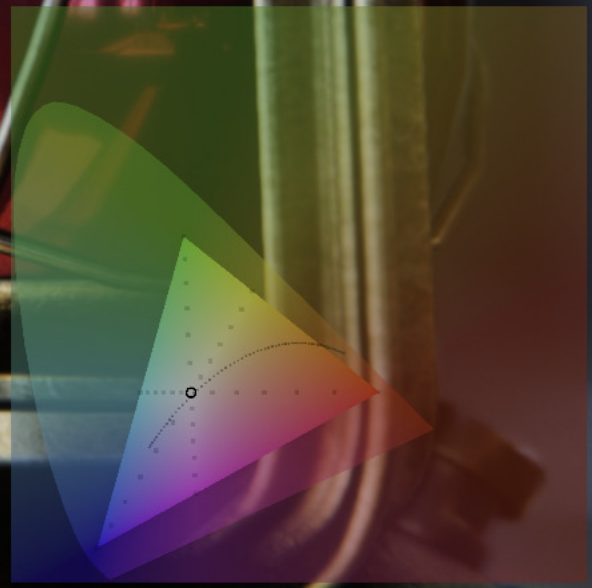


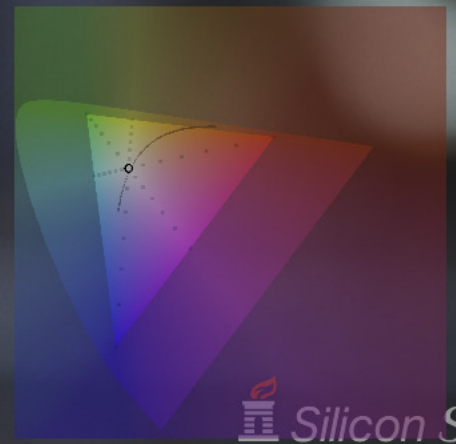






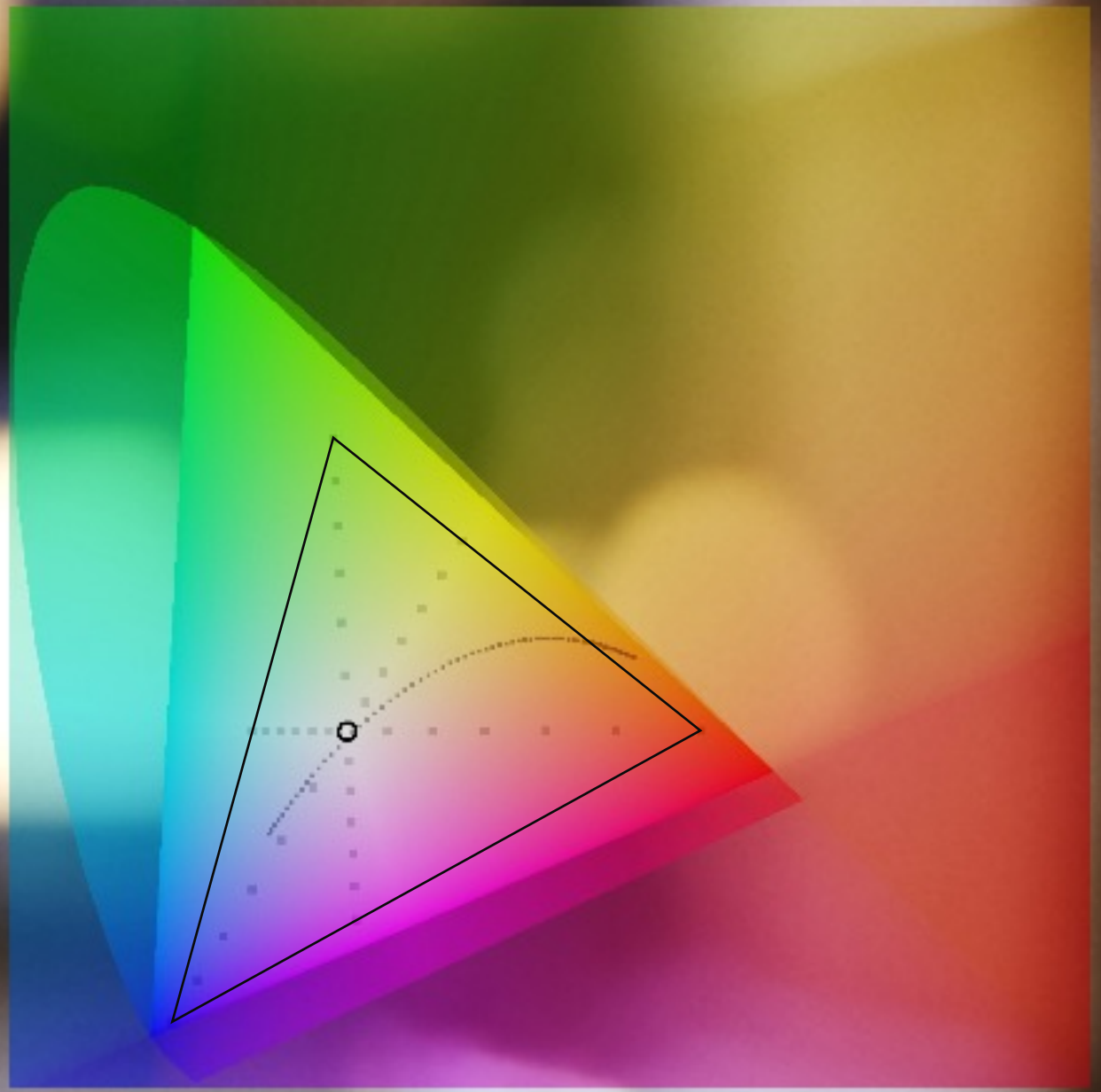


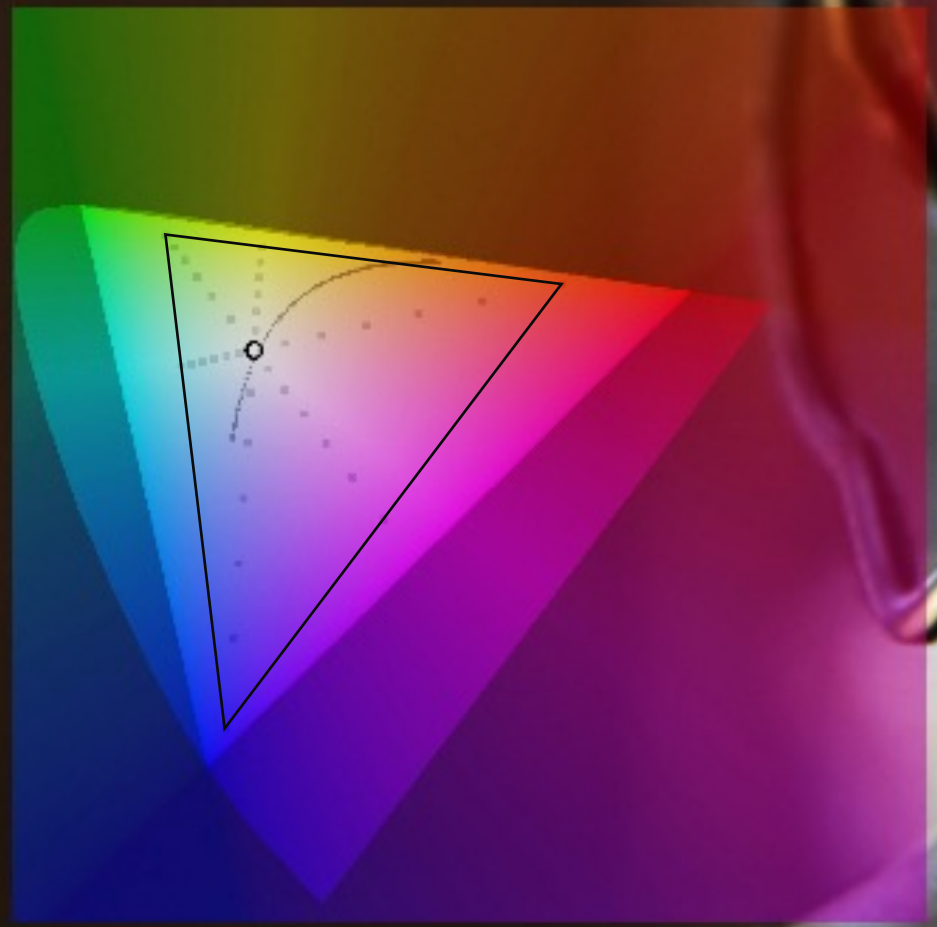




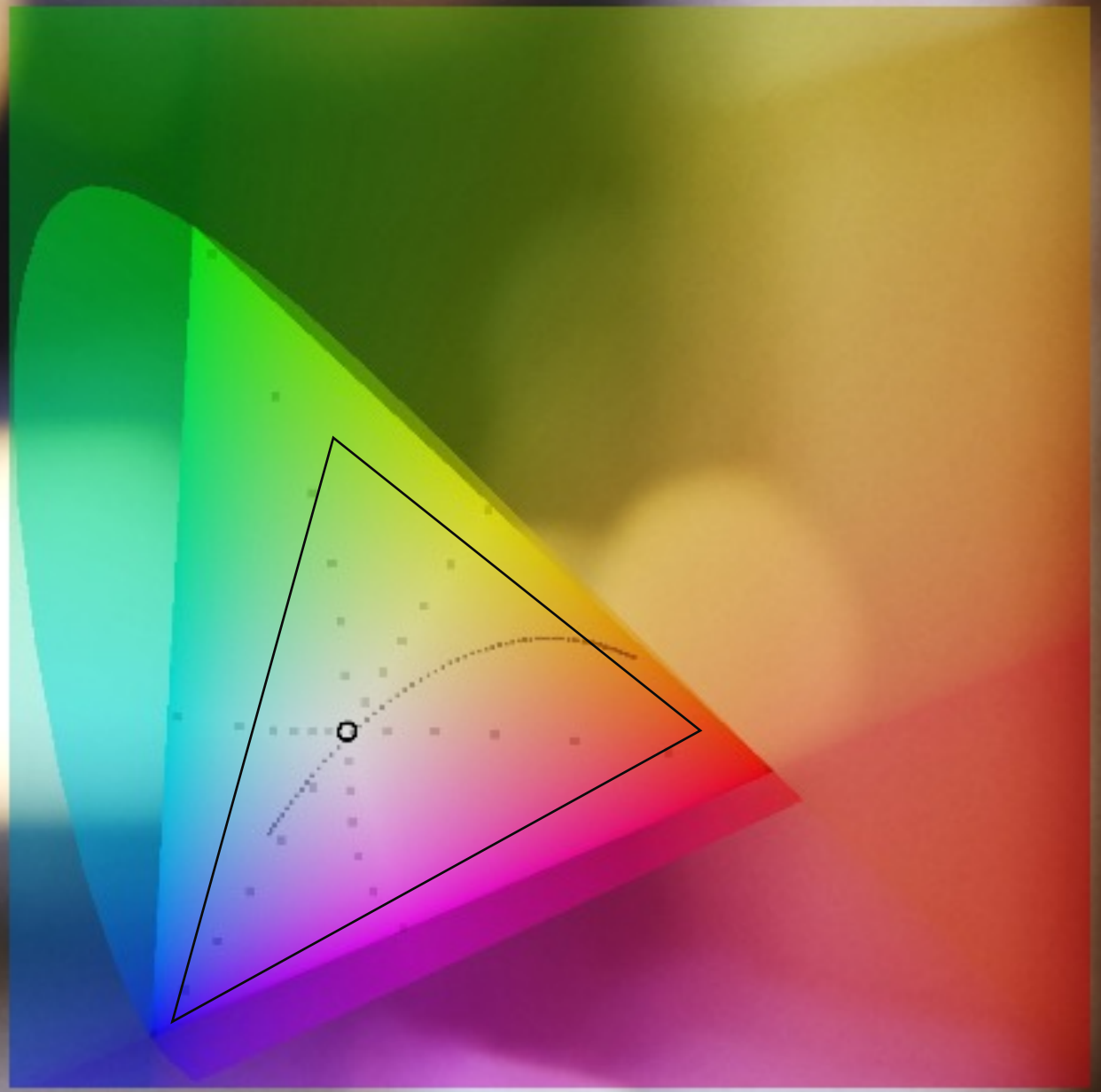
Silicon Studio
Mizuchi

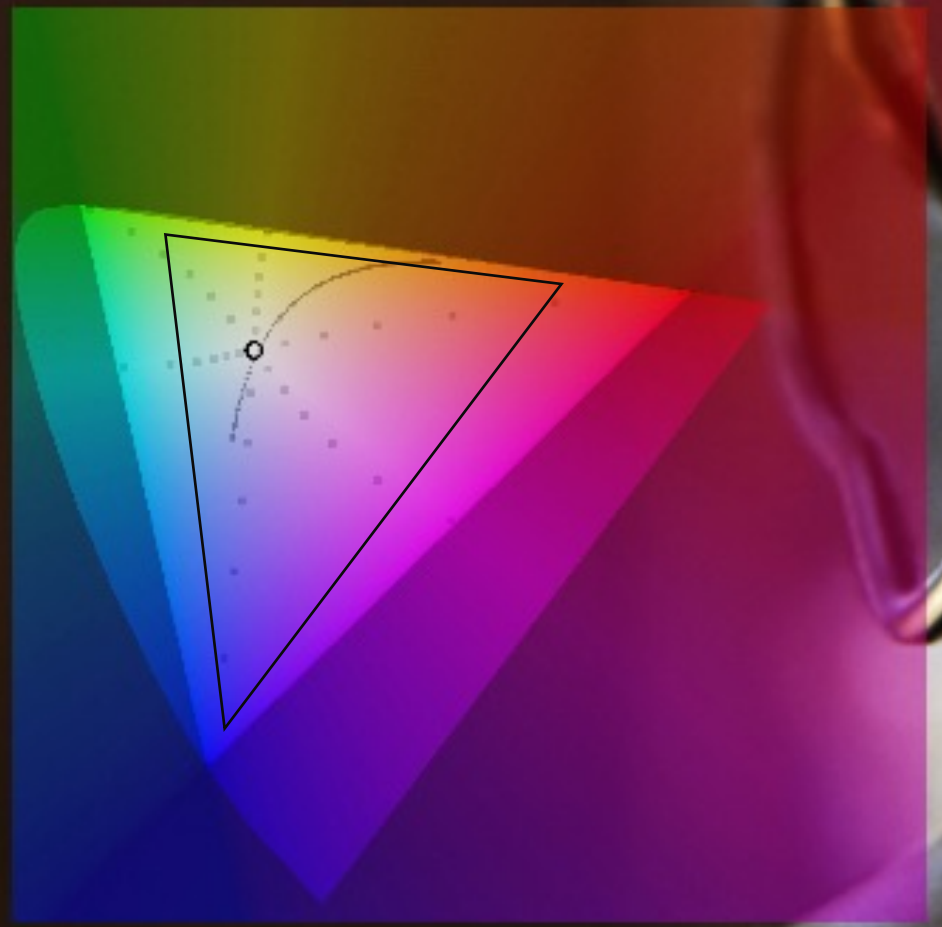
BT.709領域のみ使用





# BT.709からBT.2020へ拡張





# さまざまな色域間で汎用的に利用可能

- 色域を拡張する方向にも圧縮する方向にも利用可能
  - 一般的なRGB色空間であればほぼ適用可能
    - 色相が大きくずれている色域同士では難しい
  - それぞれパラメタの調整は必要

- 例えばBT.2020ベースでレンダリングした場合
  - 逆にSDR出力時に彩度の高い領域を滑らかに圧縮／飽和させる
    - 不自然な彩度のクリッピング(色飽和)を回避できる
  - 彩度の低い領域は変化しない
    - 全体の絵作りの印象は変化しない


©Silicon Studio Corp., all rights reserved.

# 色域圧縮マッピングの例

- BT.2020でレンダリング
- BT.709に出力

- 色域マッピング無効／有効比較

Silicon Studio ©Silicon Studio Corp., all rights reserved.

BT.2020からBT.709に制限
色域マッピング無効
BT.709領域外はクリップ

掲載の画像は権利上の理由などにより現在はご覧いただけません

BT.2020からBT.709に制限
色域マッピング有効
BT.709領域外を圧縮

掲載の画像は権利上の理由などにより現在はご覧いただけません

BT.2020からBT.709に制限
色域マッピング無効
BT.709領域外はクリップ

掲載の画像は権利上の理由などにより現在はご覧いただけません

BT.2020からBT.709に制限
色域マッピング有効
BT.709領域外を圧縮

掲載の画像は権利上の理由などにより現在はご覧いただけません

BT.2020からBT.709に制限
色域マッピング無効
BT.709領域外はクリップ

掲載の画像は権利上の理由などにより現在はご覧いただけません

BT.2020からBT.709に制限
色域マッピング有効
BT.709領域外を圧縮

掲載の画像は権利上の理由などにより現在はご覧いただけません

# 非線形圧縮の必要性？

- 条件によっては重要性が低い場合も
  - 以下のようなケース
    - BT.2020等の広色域レンダリングからBT.709等への圧縮であること
    - トーンマップをレンダリング時の広色域側で適用していること(次頁参照)
    - トーンマップが滑らかな飽和カーブであること
    - 広色域上で純色(r,g,b)=(1,1,0)のような極端なピクセル値が存在しないこと
      - 物理ベースのデータであればほとんど問題ない
      - 人為的な色では不自然な色飽和が生じる可能性がある
  - 広色域上でのトーンマップによって非線形の色域圧縮と同様の効果
    - 色域圧縮の自由度は高くない
      - トーンマップの飽和カーブ依存となるため
  - “HDR 理論と実践” CEDEC 2017 [Uchimura17] 参照


©Silicon Studio Corp., all rights reserved.

# トーンマップと色域変換の順序

- 最終出力の色域でトーンマップ
  - 本セッションで示す例
  - トーンマップの前に最終色域に変換
    - 非線形拡張や飽和も含む
- レンダリング色域でトーンマップ
  - ACES RRT等の一般的な方法
  - フレーム画像完成後に最終色域に変換
    - ODT(Output Device Transform)


©Silicon Studio Corp., all rights reserved.

# 最終出力の色域でトーンマップ

- 最終的な出力色空間の範囲を最大限に活用できる
- 広色域からの圧縮では非線形色域マッピングが必須


©Silicon Studio Corp., all rights reserved.

# レンダリング色域でトーンマップ

- 彩度の高い色が出力色空間を最大限に活かせない
  - トーンマップ段階でレンダリング色域の有効範囲内にクランプされる
- 広色域からの圧縮では色域マッピングの重要性が低い
- scRGB出力時の色域マッピングには追加のパスが必要
  - OS層でのODT変換では色域マッピングは適用できないため
  - scRGB出力については後述


©Silicon Studio Corp., all rights reserved.

# トーンマップと色域変換の順序

- 基本的には最終出力色域でのトーンマップを推奨
  - 色域変換してからトーンマップ
  - 本セッションではこちらを利用
- 条件によってはトーンマップ後の色域変換も検討
  - 一般的なワークフローに合わせたい場合
    - ACESワークフローに基づく場合など
  - 常に広色域側でトーンマップを行う選択肢も…
    - レンダリングと出力側のいずれか広い方で適用


©Silicon Studio Corp., all rights reserved.

# 色飽和の根本的な原因

- 色飽和の問題はレンダリング色域が狭くても起こる
  - RGBのいずれかがゼロ(彩度が最大)なら発生
  - そもそもBT.709は色域が狭く彩度の飽和が起こりやすい
    - 物理ベースでも容易に飽和する
      - 炎や夕方の太陽の色(2,000K未満ではB(青)がゼロとなる)
    - アーティストの色設定でも飽和しやすい
      - あえて禁止しない限り設定される可能性がある
  - 広色域から狭い色域へ変換する場合だけの問題ではない
    - 従来のSDRレンダリングでもよく発生する問題


©Silicon Studio Corp., all rights reserved.

D3D11 Vsync on (1920x1080), R8G8B8A8_UNORM (MS1, Q0)
PPFX Internal: 1920x1080, R16G16B16A16_FLOAT (0x Multisample)
HARDWARE: NVIDIA GeForce GTX 780 Ti
FPS: 60.0
Press F1 for help
Toggle full screen
Change device (F2)
by Silicon Studio
BT.709/sRGBレンダリング
純色による色飽和の問題
Reset Parameters (R)
Reset Camera (H)
Next Environment (L)

# 色飽和の軽減方法

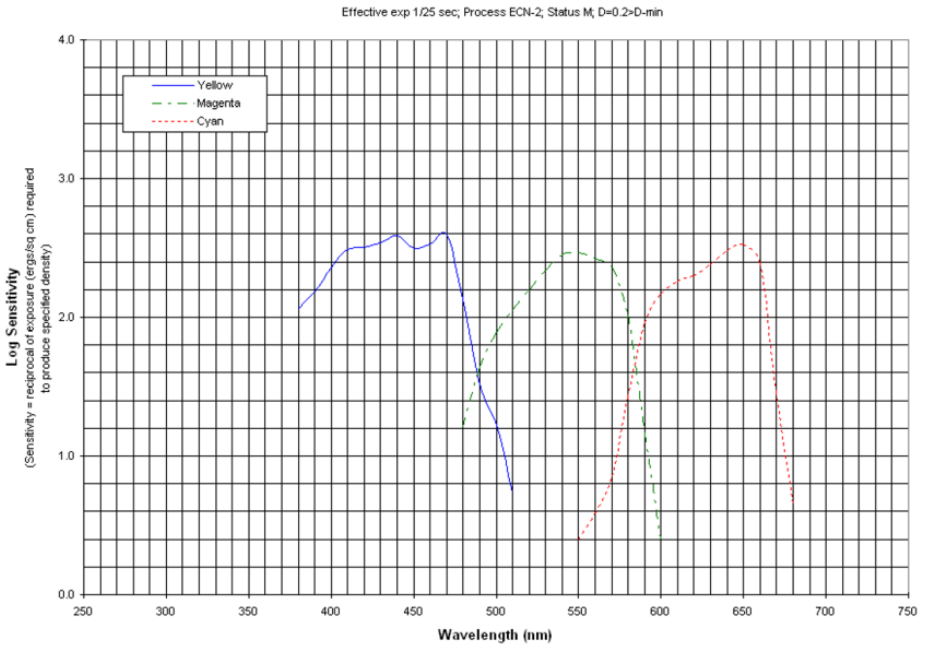
- 写真では色の不自然な飽和は起きにくい
  - カメラセンサーはもっと広色域から調整して変換している
    - ただし色設定を高彩度モードにした場合などはよく飽和する
  - フィルムの感度は他の波長にも反応するようになっている
    - フィルムの分光感度曲線参照


©Silicon Studio Corp., all rights reserved.

# フィルムの特性グラフ



- 分光感度曲線
  - 波長における、各感色層の感度を表した対数グラフ
  - ある一定の濃度の時のグラフなので、実際にはこのグラフが濃度ごとにあるのが理想
  - 一つの感色層は単純に一つの波長に反応するわけではない
    - 例えばシアンの感色層は赤にだけ反応するわけではなく、緑や青にも多少反応する
    - つまり最終的には非常に強い光はすべて白くなることを示している
  - 色の鮮やかさや肌色のコントロールのためある程度オーバーラップさせている






"レンダリストのためのカメラ（光学）理論とポストエフェクト"
CEDEC 2007 [Gotanda07] より引用

Silicon Studio
©Silicon Studio Corp., all rights reserved.

# 色飽和の軽減方法

- レンダリングでも分光感度のオーバーラップを近似
  - トーンマップ前に感度をオーバーラップさせる
    - RGBいずれかの値がゼロとなるピクセルが消滅
  - 行列による線形変換だけで簡単に調整できる


©Silicon Studio Corp., all rights reserved.

# 線形変換を利用した色飽和の軽減

- 感度のオーバーラップ近似行列の例
  - 例えば計4%程度を他のチャネルから感光させる場合

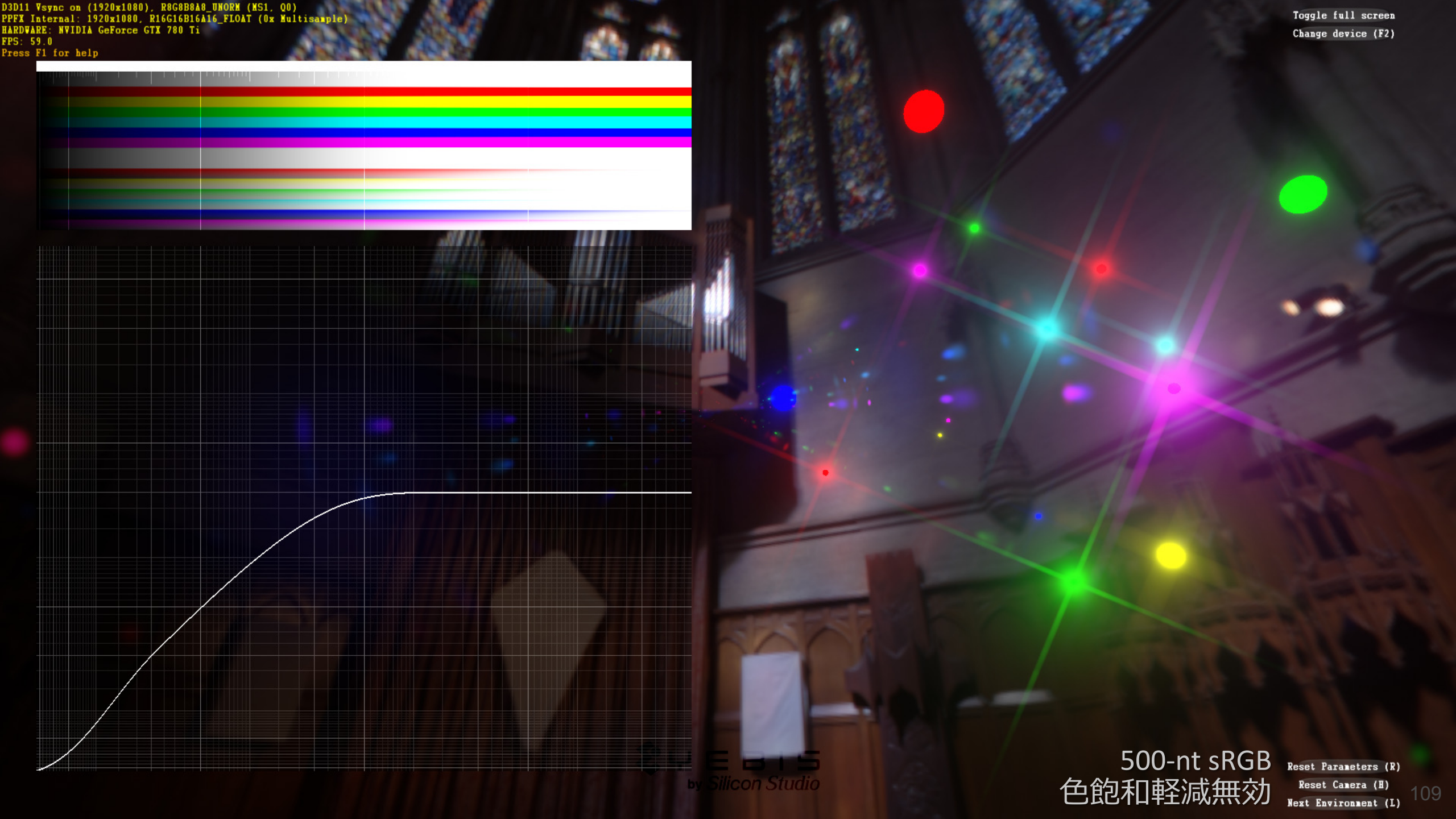
$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} 0.960 & 0.038 & 0.002 \\ 0.020 & 0.960 & 0.020 \\ 0.002 & 0.038 & 0.960 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

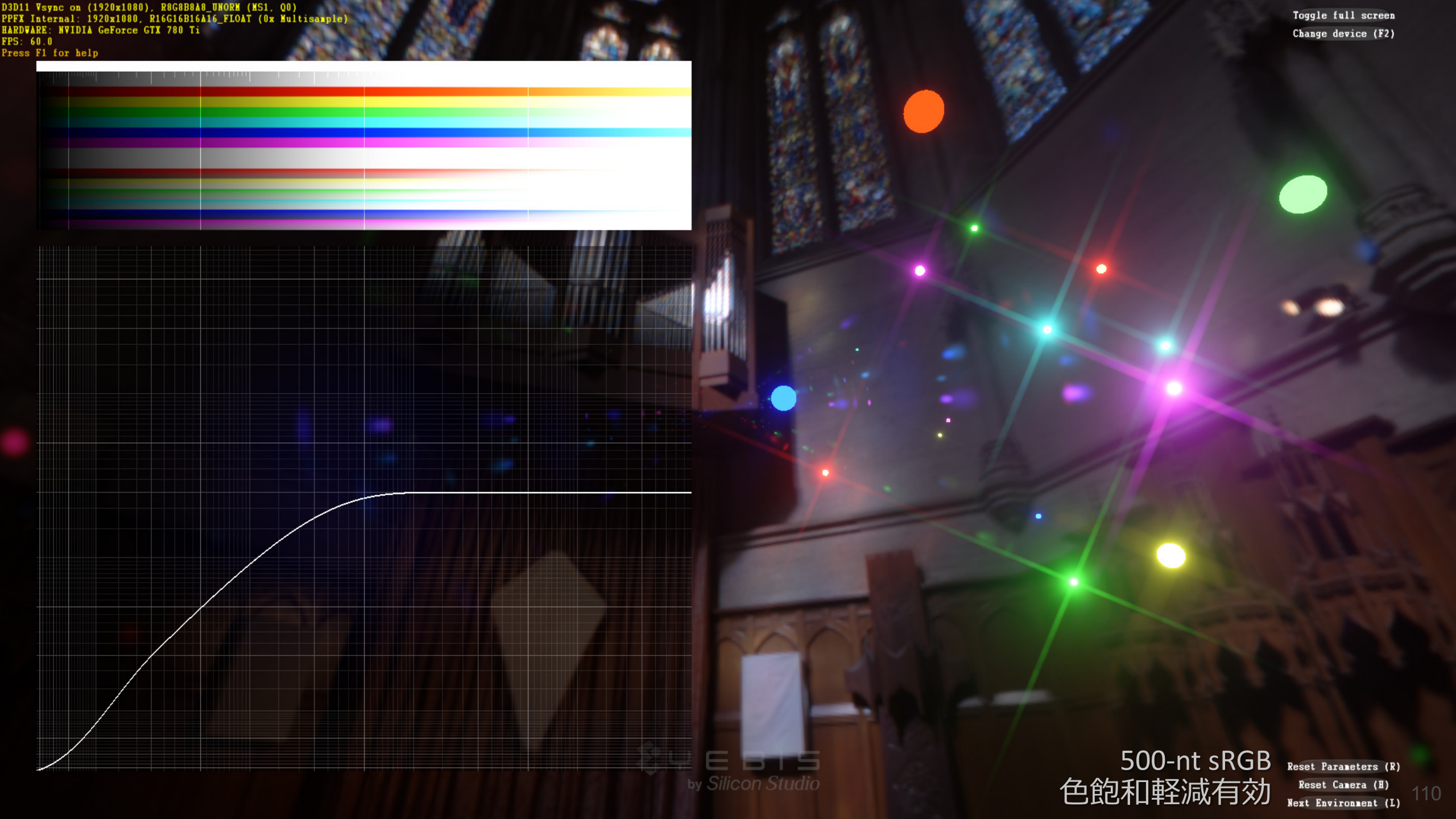
  - あまり他チャネルの感度を高くし過ぎないように注意
    - 全体的な彩度低下に繋がる
    - グレアソースだけを行列変換する方法も有効

Silicon Studio ©Silicon Studio Corp., all rights reserved.

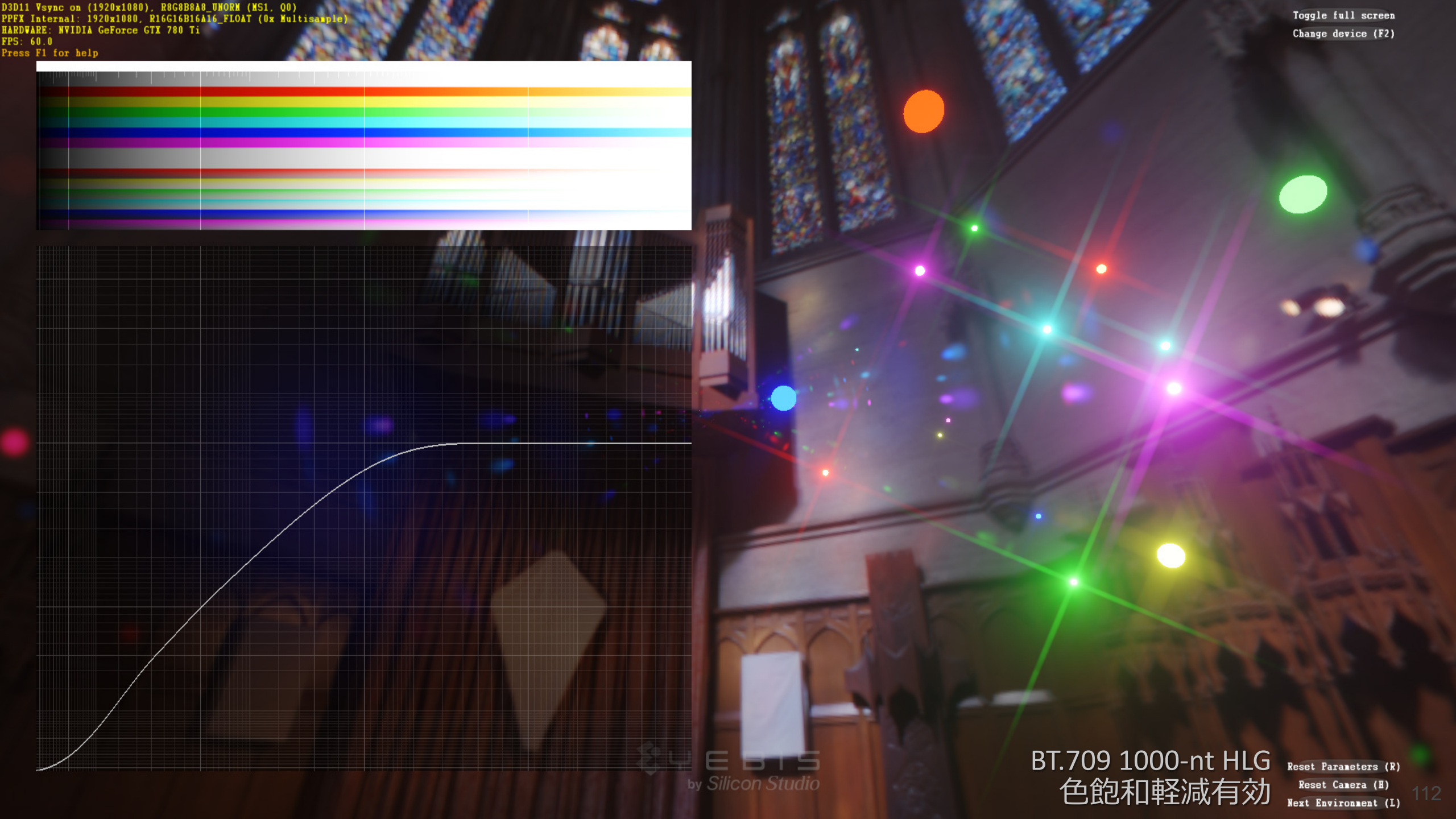
D3D11 Vsync on (1920x1080), R8G8B8A8_UNORM (MS1, Q0)
PPFX Internal: 1920x1080, R16G16B16A16_FLOAT (0x Multisample)
HARDWARE: NVIDIA GeForce GTX 780 Ti
FPS: 60.0
Press F1 for help
Toggle full screen
Change device (F2)
by Silicon Studio
100-nt sRGB
色飽和軽減無効
Reset Parameters (R)
Reset Camera (H)
Next Environment (L)

D3D11 Vsync on (1920x1080), R8G8B8A8_UNORM (MS1, Q0)
PPFX Internal: 1920x1080, R16G16B16A16_FLOAT (0x Multisample)
HARDWARE: NVIDIA GeForce GTX 780 Ti
FPS: 60.0
Press F1 for help
Toggle full screen
Change device (F2)
by Silicon Studio
100-nt sRGB
色飽和軽減有効
Reset Parameters (R)
Reset Camera (H)
Next Environment (L)


D3D11 Vsync on (1920x1080), R8G8B8A8_UNORM (MS1, Q0)
PPFX Internal: 1920x1080, R16G16B16A16_FLOAT (0x Multisample)
HARDWARE: NVIDIA GeForce GTX 780 Ti
FPS: 59.0
Press F1 for help
Toggle full screen
Change device (F2)
by Silicon Studio
500-nt sRGB
色飽和軽減無効
Reset Parameters (R)
Reset Camera (H)
Next Environment (L)


D3D11 Vsync on (1920x1080), R8G8B8A8_UNORM (MS1, Q0)
PPFX Internal: 1920x1080, R16G16B16A16_FLOAT (0x Multisample)
HARDWARE: NVIDIA GeForce GTX 780 Ti
FPS: 60.0
Press F1 for help
Toggle full screen
Change device (F2)
by Silicon Studio
500-nt sRGB
色飽和軽減有効
Reset Parameters (R)
Reset Camera (H)
Next Environment (L)

D3D11 Vsync on (1920x1080), R8G8B8A8_UNORM (MS1, Q0)
PPFX Internal: 1920x1080, R16G16B16A16_FLOAT (0x Multisample)
HARDWARE: NVIDIA GeForce GTX 780 Ti
FPS: 60.0
Press F1 for help
Toggle full screen
Change device (F2)
by Silicon Studio
BT.709 1000-nt HLG
色飽和軽減無効
Reset Parameters (R)
Reset Camera (H)
Next Environment (L)


D3D11 Vsync on (1920x1080), R8G8B8A8_UNORM (MS1, Q0)
PPFX Internal: 1920x1080, R16G16B16A16_FLOAT (0x Multisample)
HARDWARE: NVIDIA GeForce GTX 780 Ti
FPS: 60.0
Press F1 for help
Toggle full screen
Change device (F2)
by Silicon Studio
BT.709 1000-nt HLG
色飽和軽減有効
Reset Parameters (R)
Reset Camera (H)
Next Environment (L)

# 前提として押さえておきたいこと

- 白レベル(Paper White Level)の調整
- SDR時のシステムガンマとの互換性
- 色域変換と色域マッピング
- ビデオ出力設定の仕様


©Silicon Studio Corp., all rights reserved.

# ビデオ出力設定の仕様

- HDR10等
  - BT.2020 PQ非線形色空間での書き出し
    - RGB10_A2/RGBA16_UNORMフォーマットに出力
  - scRGB 線形色空間での書き出し
    - RGBA16_FLOATフォーマットに出力
- HLG
  - 現状ではリアルタイムコンテンツにおける対応無し
    - 対応する場合はRGB10_A2/RGBA16_UNORMに出力
- Dolby Vision
  - RGBA8_UNORMフォーマットに出力
    - 内部データとしては12-bit YCbCr 4:2:2
- etc.


©Silicon Studio Corp., all rights reserved.

# ビデオ出力設定の仕様

- 詳細は各プラットフォーム等のドキュメント参照
- 重要な点：
  - プラットフォームによって利用できる方式が異なる
  - 互換性のためには複数の方式への対応が必要

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 基本的なレンダリングフローの検討

本セッションの目的
色空間
前提として押さえておきたいこと
基本的なレンダリングフローの検討
その他の考慮すべきこと
互換性重視レンダリングフローの構築
まとめ

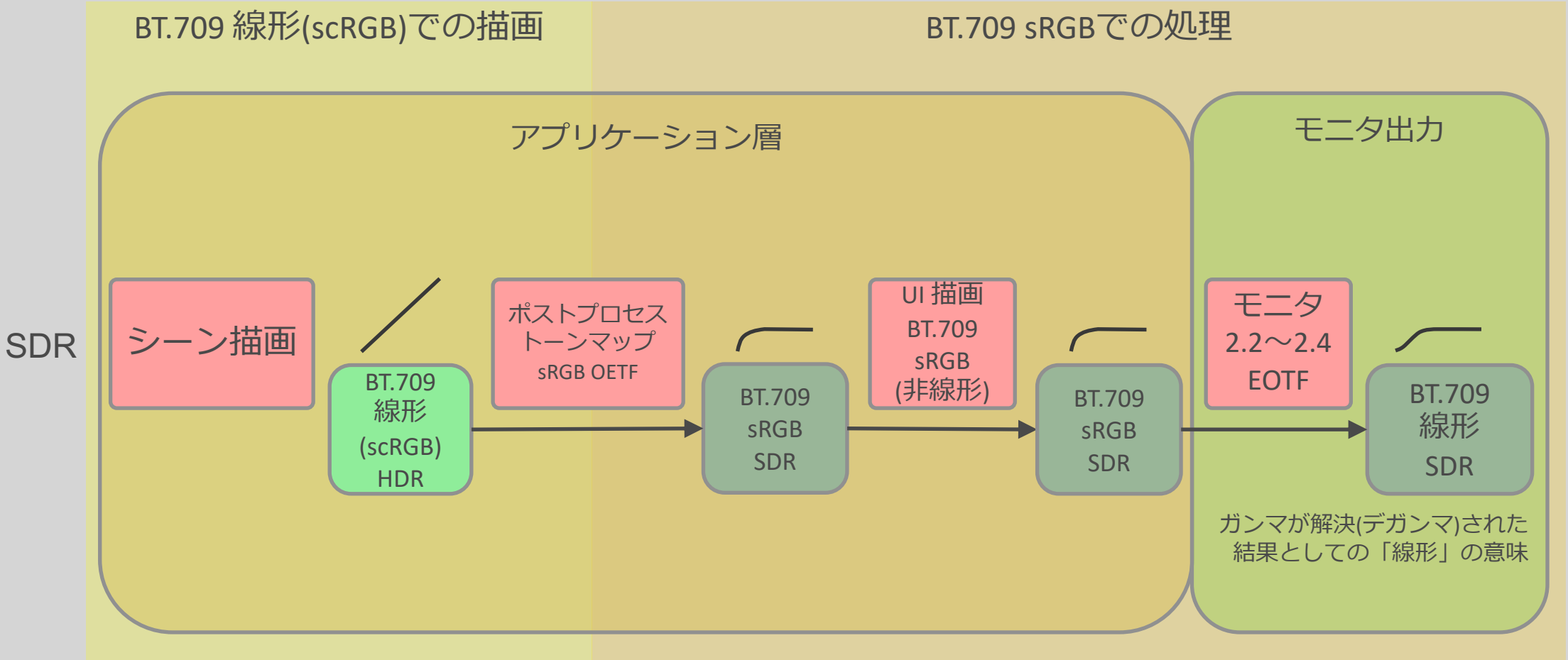
Silicon Studio
©Silicon Studio Corp. all rights reserved!

# 基本的なレンダリングフローの検討

- さまざまな条件でのフロー例を検討
  - プラットフォームによるビデオ出力設定のタイプ
  - OETFのタイミング
    - パフォーマンスと正しさや互換性の考察
- 前提
  - ビデオ出力モードは適切に設定されている
    - HDR非対応時にはSDR出力が設定されている
    - HDR対応テレビ接続時には適切なHDR出力が設定されている


©Silicon Studio Corp., all rights reserved.

# 典型的な線形HDRレンダリング SDR出力フロー




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# HDR OETF対応レンダリングフロー

- 大きく2種類の適用方法
  - 即時OETF適用フロー
    - 一般的なトーンマップパスと同時にOETFも適用
  - 遅延OETF適用フロー
    - トーンマップパスとは別にUI描画後にOETFを適用
  - ビデオ出力方式やパフォーマンスで選択
    - 方式やプラットフォームで選択肢が制限される


©Silicon Studio Corp., all rights reserved.

# 即時OETF適用

- トーンマップ＆OETF→UI描画
- 従来のSDRとほぼ同じフロー
  - トーンマップと同時にOETF(いわゆるガンマ補正)を行う
  - シンプルで理解しやすい

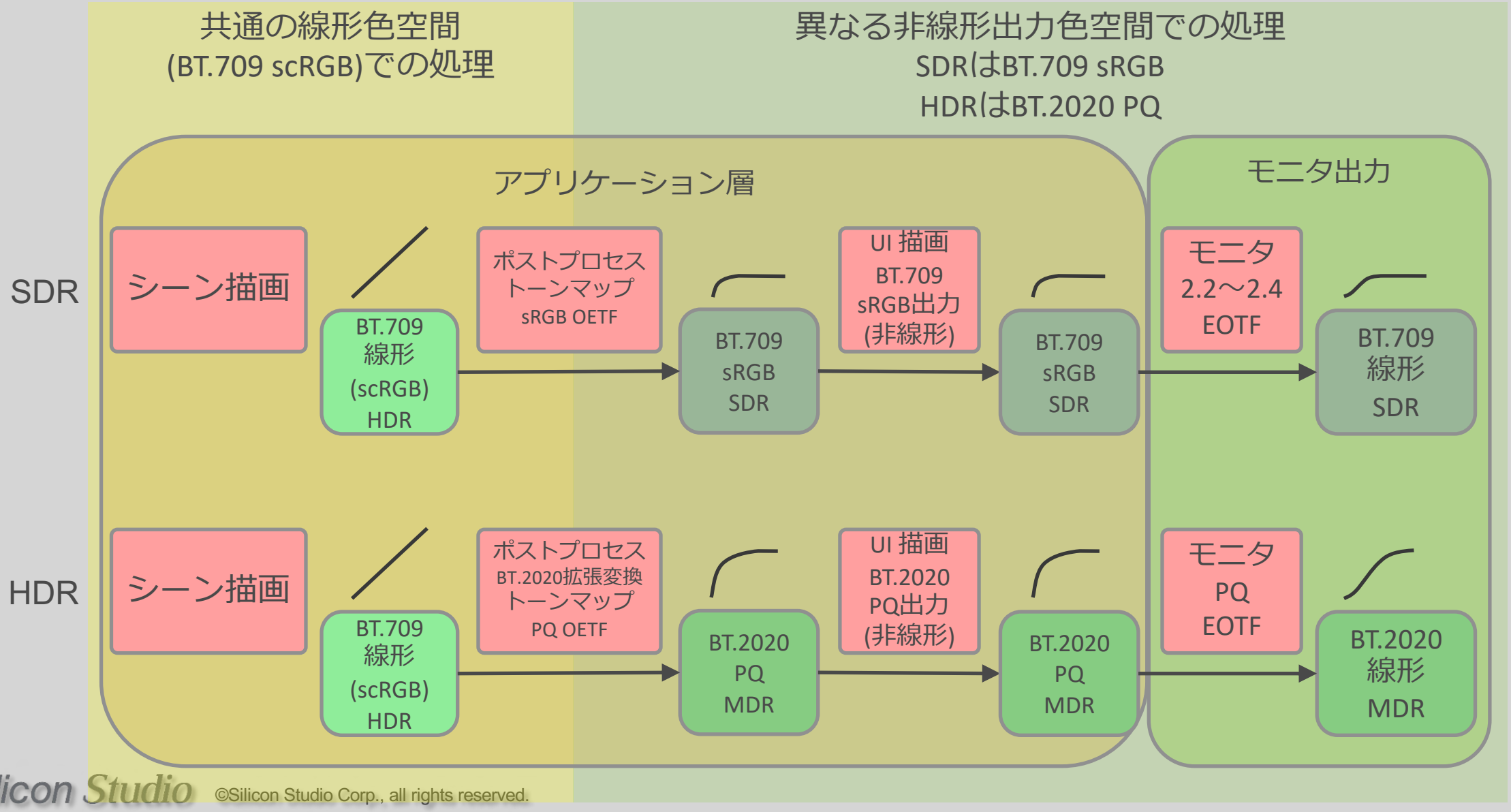
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 即時OETF適用フロー例

- HDR10等
  - BT.2020 PQ非線形色空間での書き出し
    - RGB10_A2/RGBA16_UNORMフォーマットに出力
  - scRGB 線形色空間での書き出し
    - RGBA16_FLOATフォーマットに出力
- HLG
  - 現状ではリアルタイムコンテンツにおける対応無し
    - 対応する場合はRGB10_A2/RGBA16_UNORMに出力
- Dolby Vision
  - RGBA8_UNORMフォーマットに出力
    - 内部データとしては12-bit YCbCr 4:2:2
- etc.

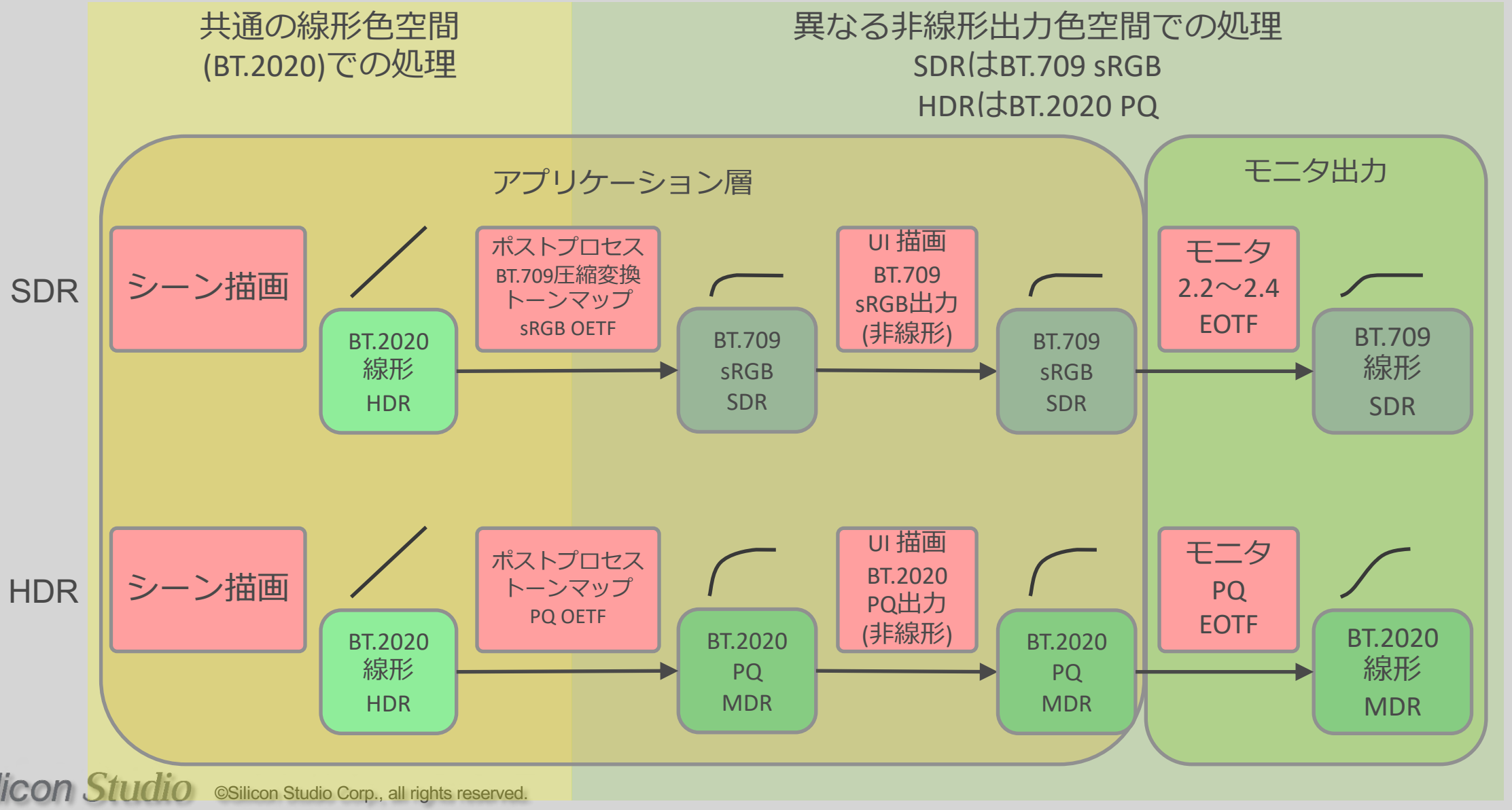

©Silicon Studio Corp., all rights reserved.

# BT.709ベース(UI描画は出力色空間別)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# BT.2020ベース(UI描画は出力色空間別)



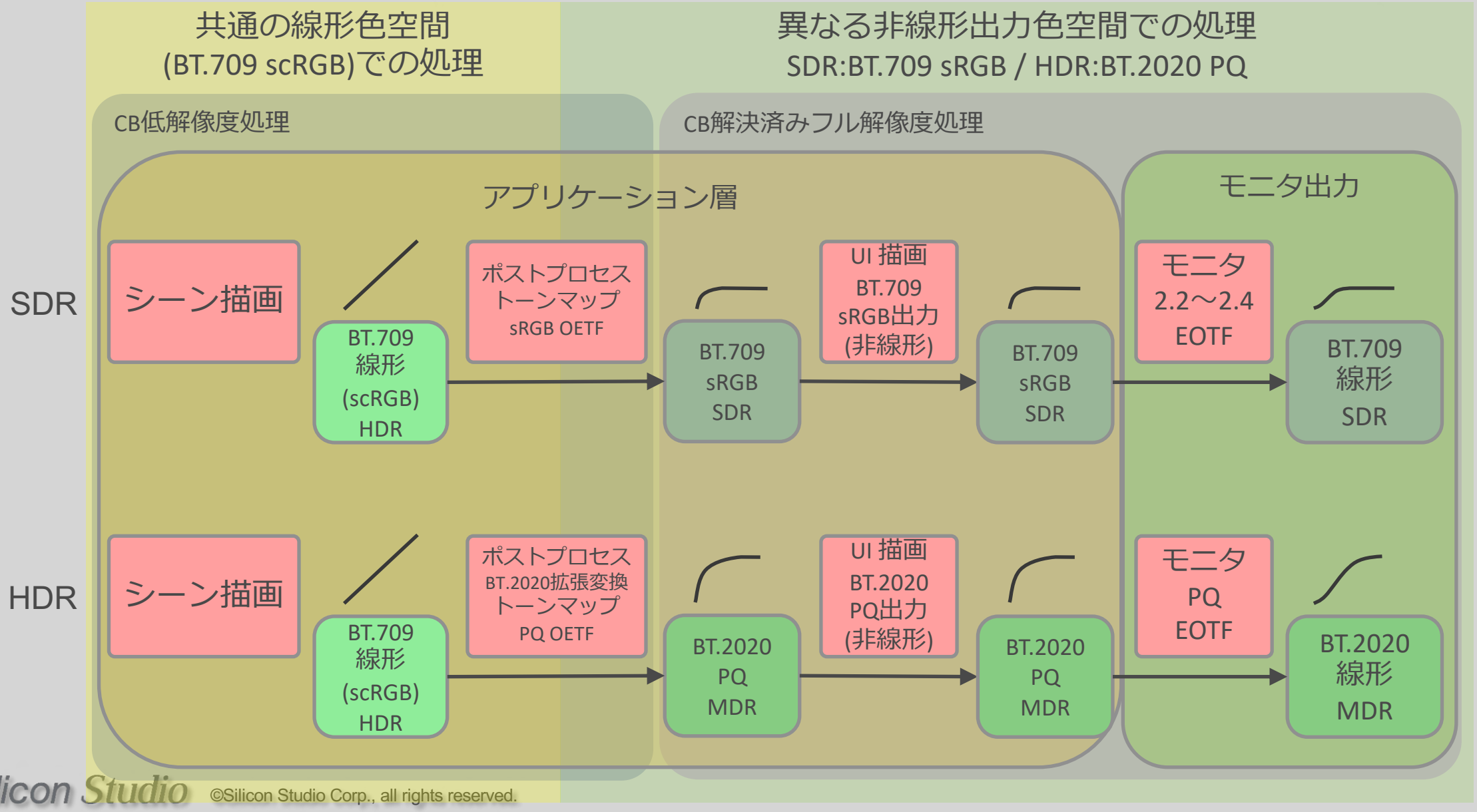

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 即時OETFにおける留意点

- UI描画を非線形な出力色空間で行う必要あり
  - SDR時は従来通りsRGB色空間で描画
  - HDR時はBT.2020 PQ(ST 2084)色空間で描画
    - BT.709色域素材の場合は変換してから出力
    - ブレンドやエッジアンチエイリアスが不正確
      - 非線形なOETF空間でブレンドされるため
- チェッカーボード(CB)レンダリングでOETFの効率化可能
  - トーンマップ＆OETFパスを縮小解像度のまま処理できる
  - ただし、ポストプロセスも縮小解像度で実装する制限あり
    - AAやレンズディストーション、倍率色収差等は使用できない
    - エッジ情報が適切ではなくなるため

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# CB例(BT.709, アプリ層による即時OETF)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 遅延OETF適用

- トーンマップ→UI描画→OETF
- 理論的には即時OETFよりもこちらの方が適切
  - OETFは本来ODT(Output Device Transform)としての役割
- アプリ側のレンダリングパイプラインを共通化
  - すべてのパスで共通の色域(あるいは色空間)を利用可能
- 追加パスが必要
  - トーンマップパスとは別にフル解像度のOETF変換パスが必須

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 2種類の遅延OETF

- アプリケーション層によるODT変換
  - アプリ側でモードに応じて適切な色空間に変換
    - sRGBあるいはBT.2020 PQ色空間
- OS／ドライバ層による自動ODT変換
  - アプリ側は常にBT.709線形(scRGB)フォーマットへ出力
  - OS／ドライバ層がモードに応じて自動的にODT変換

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# アプリ側遅延OETF適用フロー

- アプリケーション層によるODT変換
- HDR方式によっては事実上必須
  - フレームが完成してから最後にまとめて変換する必要がある場合
  - 例えばDolby Vision
    - OETF後にRGBA8888バッファに特殊なエンコードが必要
      - UI等の描画中にエンコードは現実的ではない
      - ブレンド等も利用できない
  - 例えばHDRとSDRを両方出力しなければならない環境
    - UI描画前にOETFを適用するとUI描画も2画面出力しなければならない
    - →パフォーマンスにもよるが最後にまとめて変換する方が効率的な場合

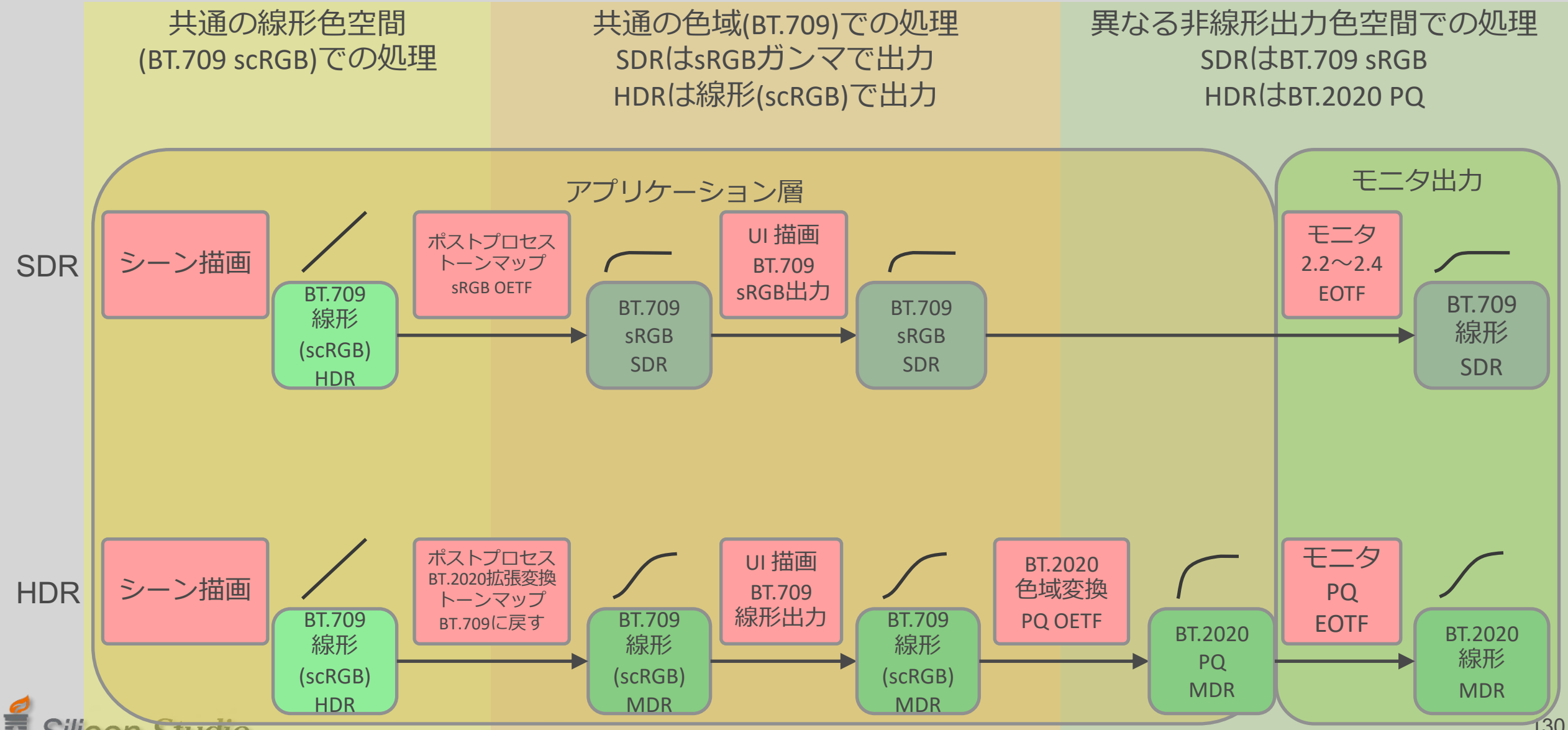
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# アプリ側遅延OETF適用フロー

- HDR10等
  - BT.2020 PQ非線形色空間での書き出し
    - RGB10_A2/RGBA16_UNORMフォーマットに出力
  - scRGB 線形色空間での書き出し
    - RGBA16_FLOATフォーマットに出力
- HLG
  - 現状ではリアルタイムコンテンツにおける対応無し
    - 対応する場合はRGB10_A2/RGBA16_UNORMに出力
- Dolby Vision
  - RGBA8_UNORMフォーマットに出力
    - 内部データとしては12-bit YCbCr 4:2:2
- etc.

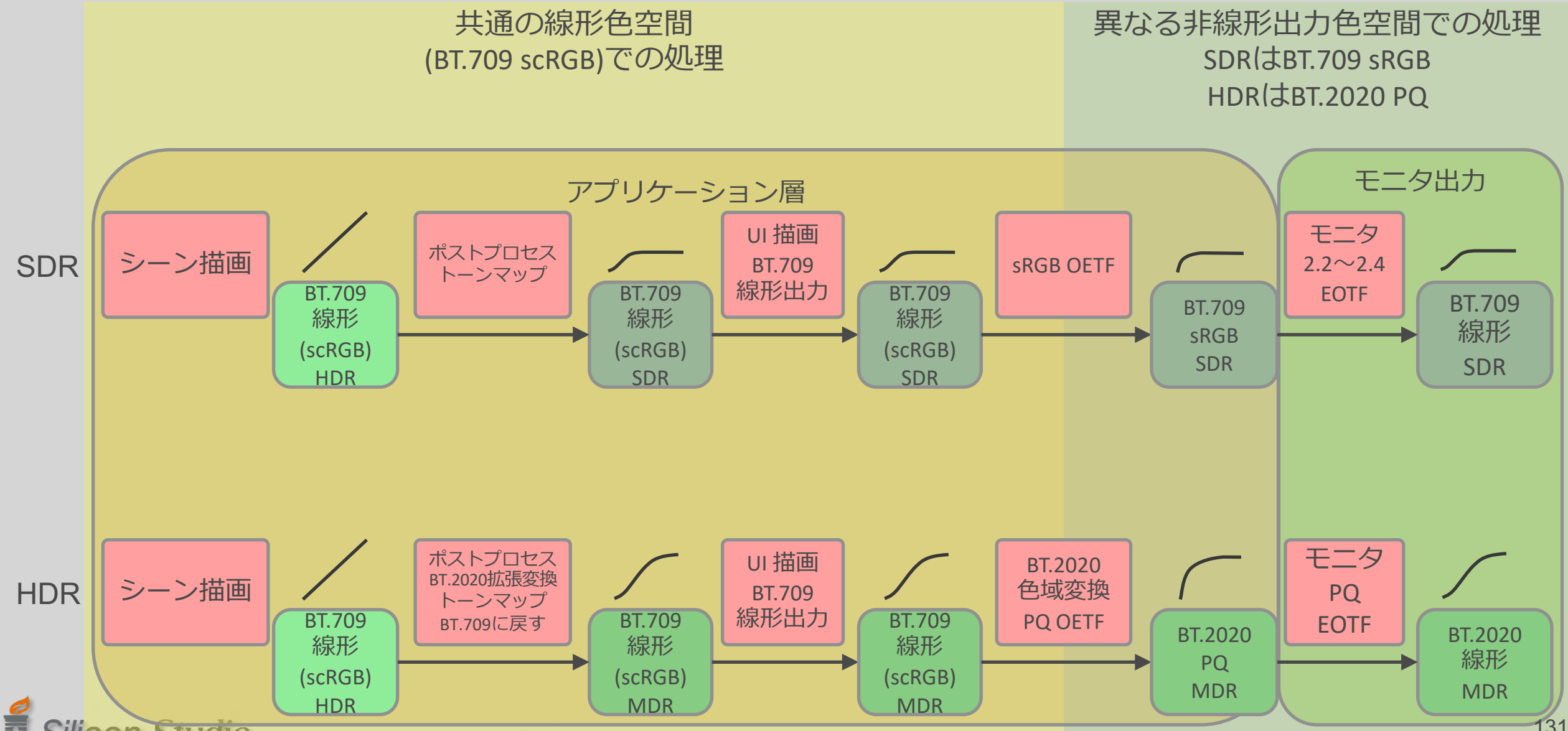
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# BT.709ベース(UI描画はBT.709でOETFだけ別)



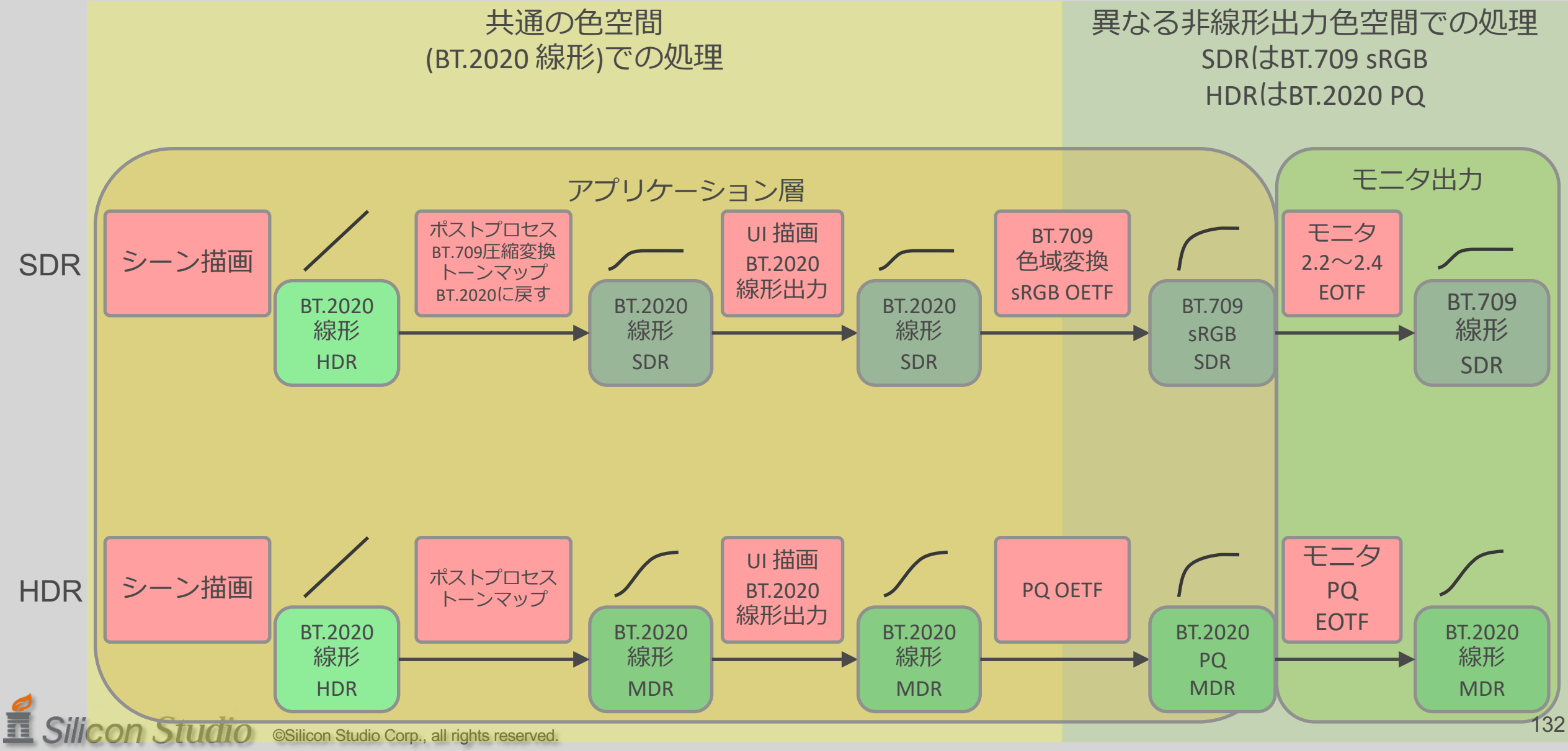

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# BT.709ベース(UI描画もBT.709線形で共通)




Silicon Studio
©Silicon Studio Corp., all rights reserved.

# BT.2020ベース(UI描画もBT.2020線形で共通)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OS側遅延OETF適用フロー(scRGB出力)

- OS／ドライバ層によるODT変換
  - 自動的に適切な色域／OETF変換が入る
  - アプリ側は常にBT.709 線形(scRGB)で出力する
    - HDR/SDR等の最終的な出力色空間とは無関係
    - 負の値も1.0を超える値も許されるため、BT.2020等の広色域の色情報も格納できる
  - SDR時の出力は従来通りのsRGBでも良い

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OS側遅延OETF適用フロー(scRGB出力)

- OS／ドライバ層によるODT変換
- OS／ドライバ側のサポートが必須
  - DXGI (Windows 10)
  - NVAPI (NVIDIA)
    - scRGB出力しかサポートしない
  - AGS API (AMD)
  - etc.

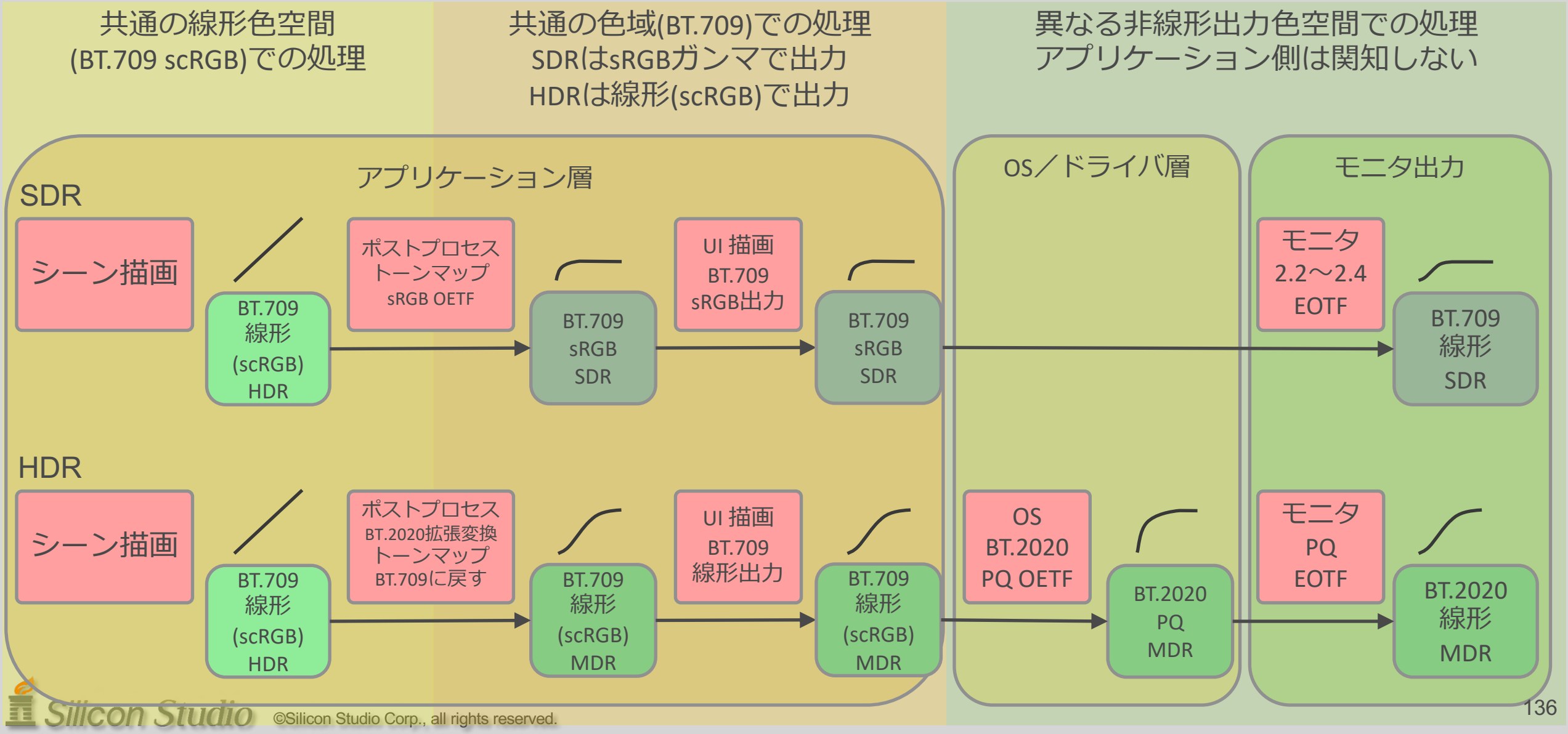
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OS側遅延OETF適用フロー(scRGB出力)

- HDR10等
  - BT.2020 PQ非線形色空間での書き出し
    - RGB10_A2/RGBA16_UNORMフォーマットに出力
  - scRGB 線形色空間での書き出し
    - RGBA16_FLOATフォーマットに出力
- HLG
  - 現状ではリアルタイムコンテンツにおける対応無し
    - 対応する場合はRGB10_A2/RGBA16_UNORMに出力
- Dolby Vision
  - RGBA8_UNORMフォーマットに出力
    - 内部データとしては12-bit YCbCr 4:2:2
- etc.

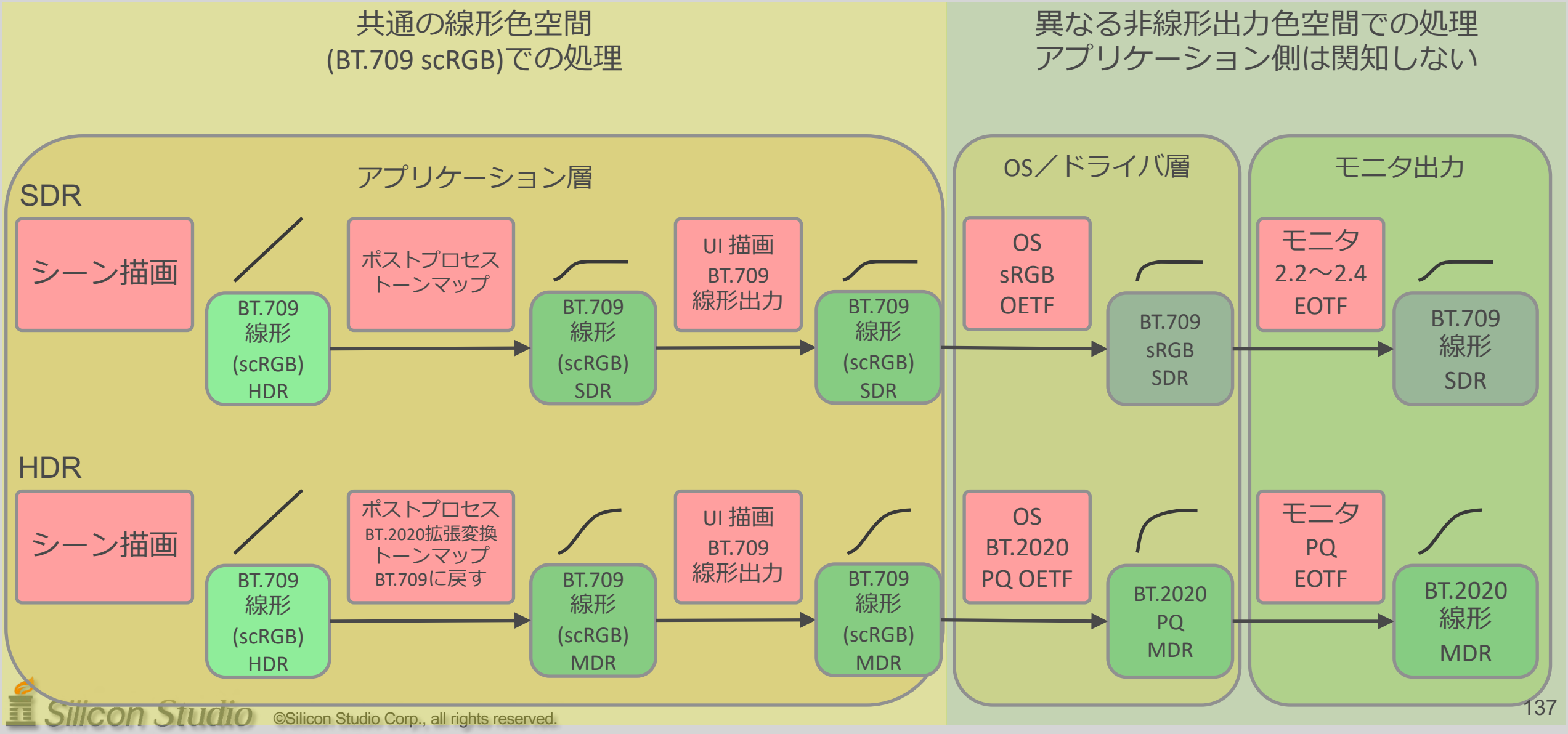
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# BT.709ベース(SDRはsRGB、 HDRはscRGB出力)



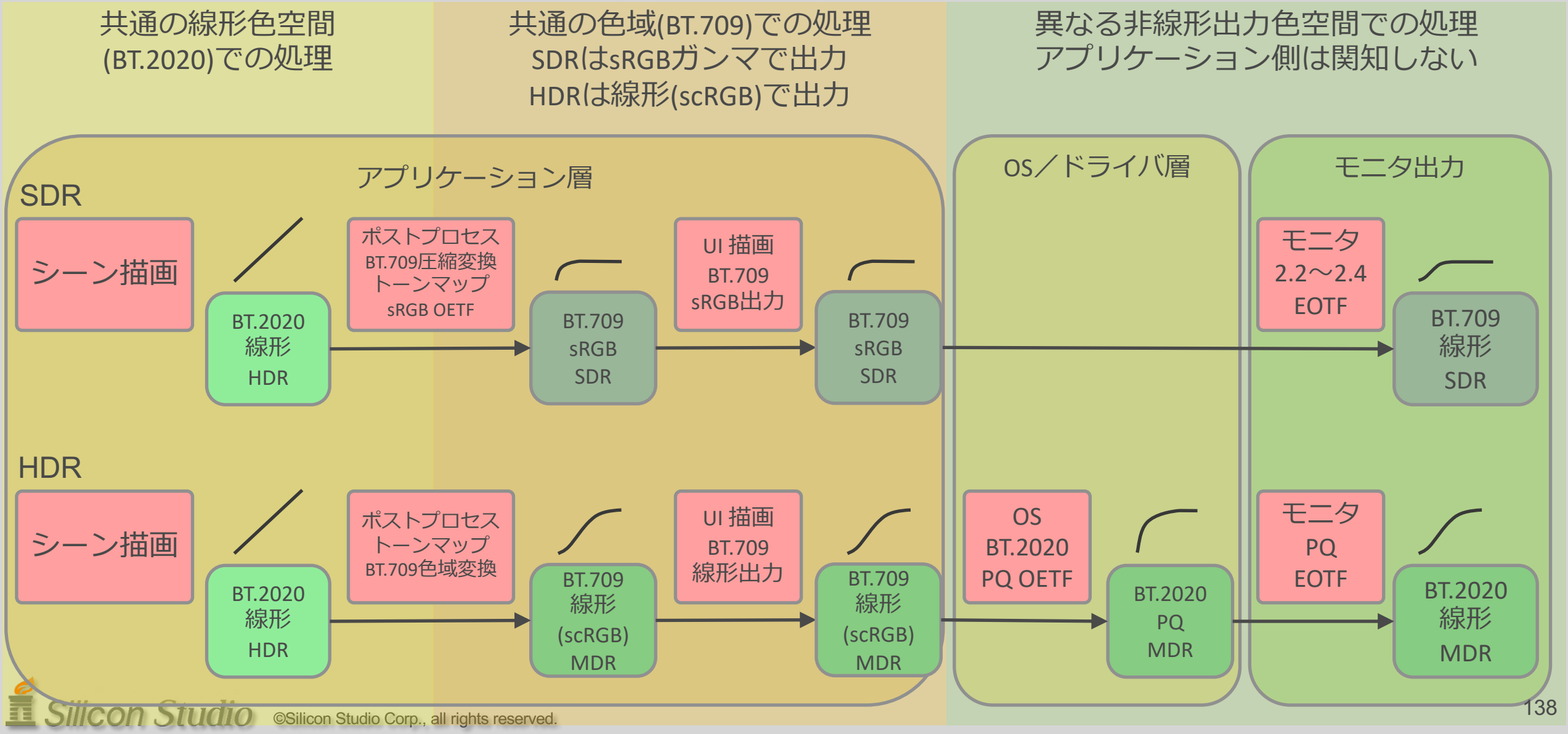

©Silicon Studio Corp., all rights reserved.

# BT.709ベース(すべてscRGBに出力)



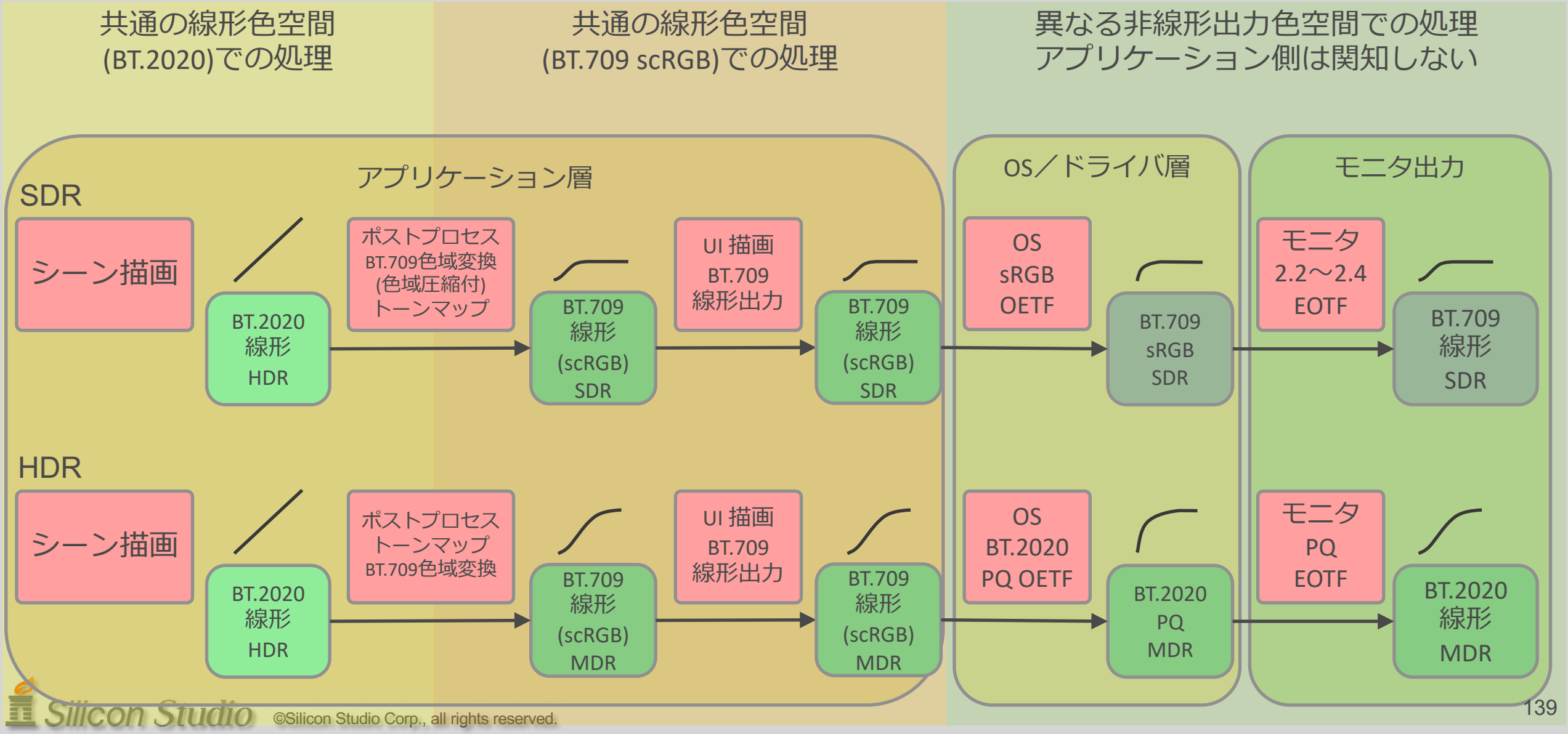

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# BT.2020ベース(SDRはsRGB、 HDRはscRGB出力)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# BT.2020ベース(UI描画はscRGB)



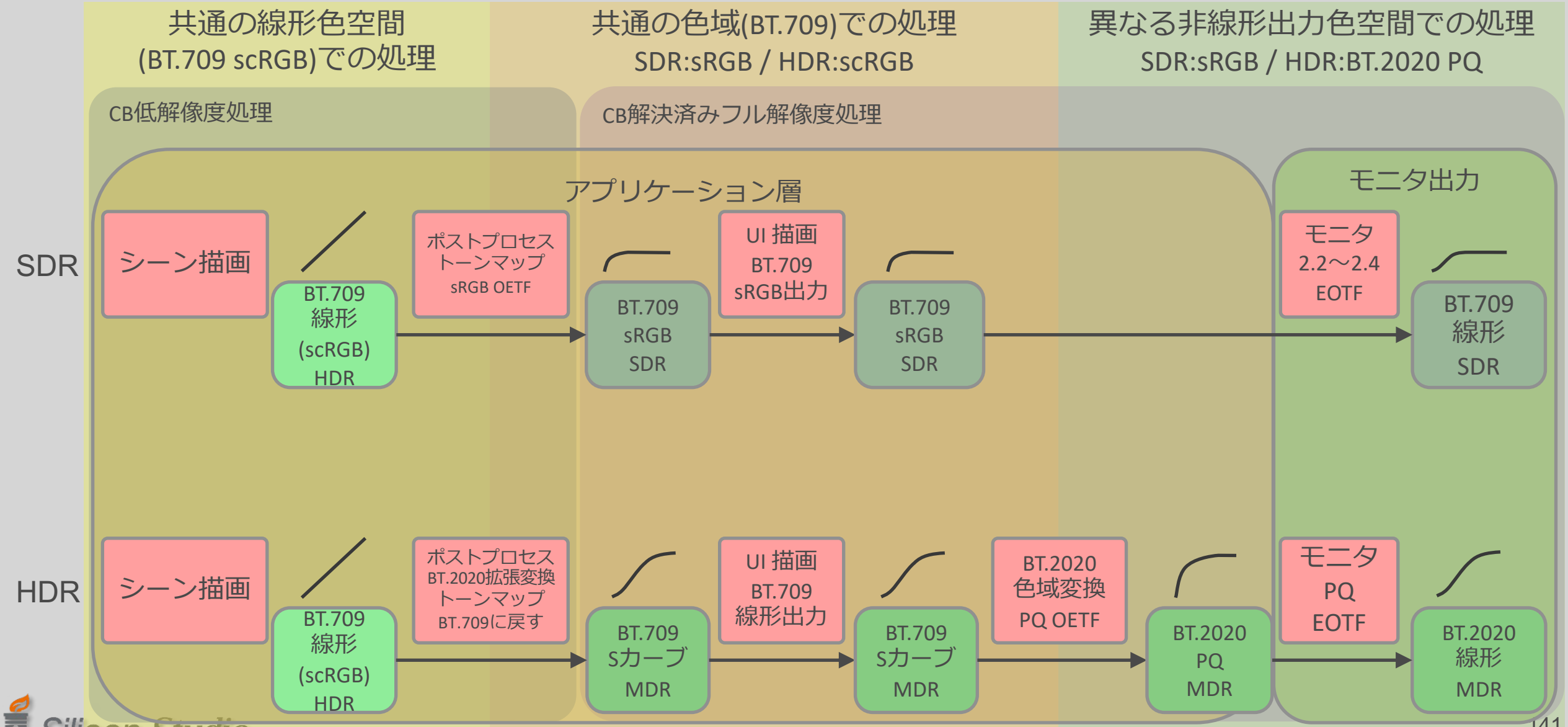

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 遅延OETF全般における留意点

- SDR時はトーンマップでsRGB出力した方が遅延OETFパスが無いため効率的
- UI描画はSDR/HDR共に同じ色空間を利用可能
  - ブレンドやエッジアンチエイリアスが正確
    - BT.2020やHDR出力時はOETF適用前の線形色空間でブレンド
    - sRGB使用時もハードウェアsRGB出力機能を利用すれば正しくブレンド可
- カラーグレーディングとしてのガンマ調整
  - 従来のSDR時に、sRGB変換と同時にガンマ調整を行っていた場合
  - SDR/HDR問わずUIをscRGB色空間で描画する場合は
    - トーンマップ後UI描画前にガンマ調整を適用する必要あり
    - ただしSDR互換OOTF適用時は、SDR OETFでガンマ調整を行う方が効率的(後述)
- CBレンダリング時にOETFパスを効率化できない
  - UI描画は解決後のフル解像度で行う必要がある
    - 必然的にUI描画後のOETF変換もフル解像度となる

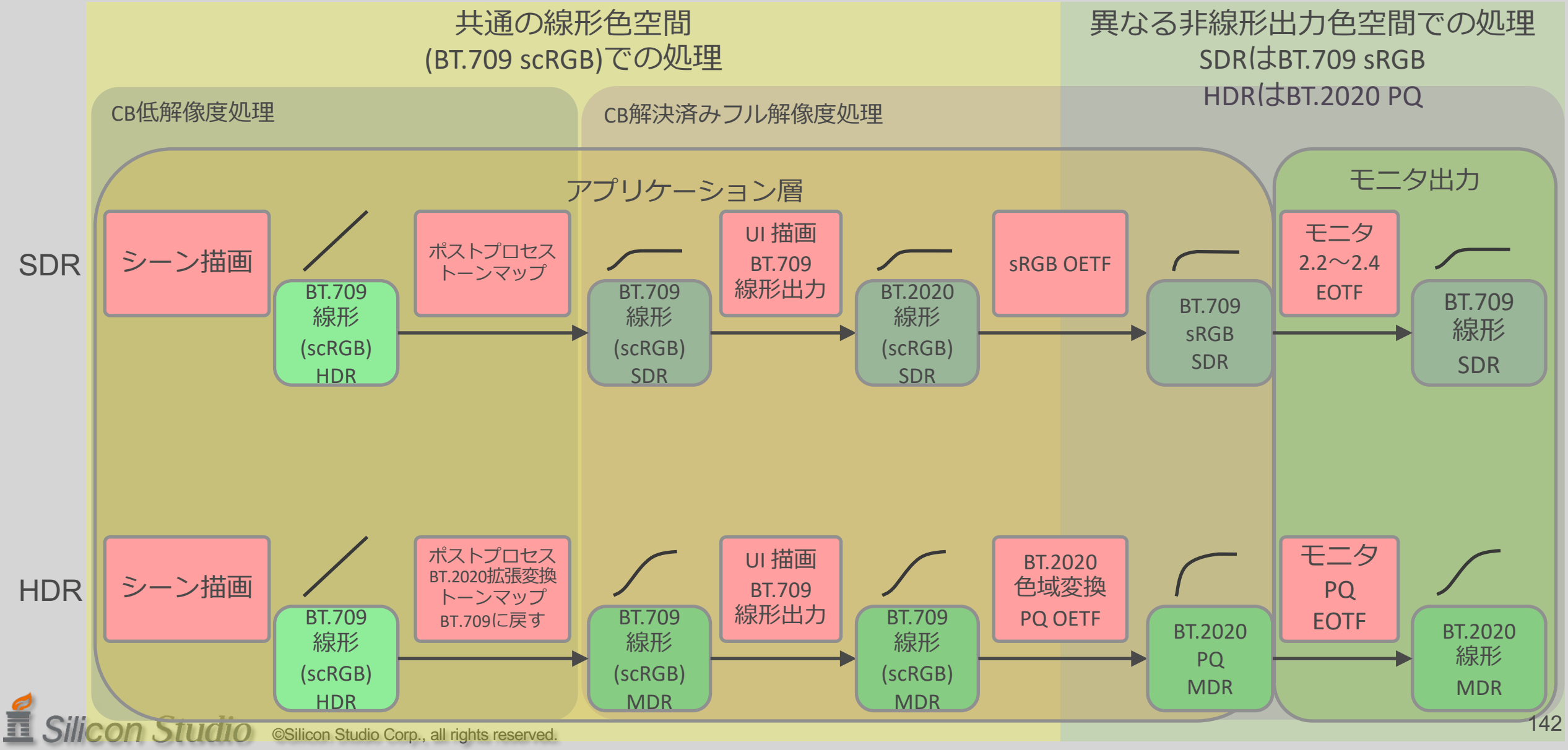
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# CB例(BT.709, SDR:即時 / HDR:アプリ遅延OETF)



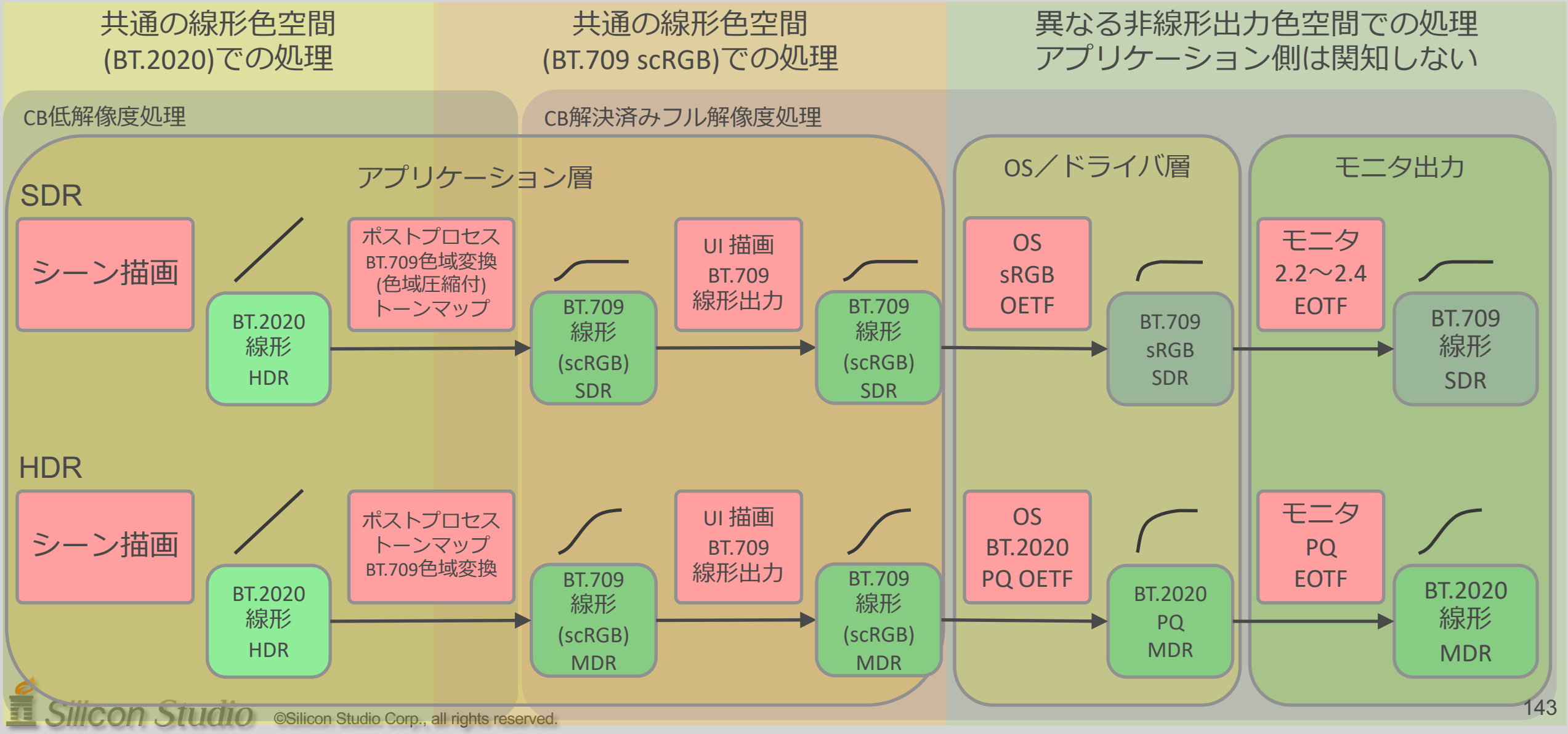

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# CB例(BT.709, アプリ層による遅延OETF)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# CB例(BT.709, OS／ドライバ層遅延OETF)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OS側遅延OETF(scRGB出力)時の留意点

- OS／ドライバ層による自動色域／OETF変換が入る
  - フル解像度を変換するコストが暗黙的に発生
- バックバッファにRGBA16_FLOATが必要
  - 広色域をscRGBに格納するには負の値が必要になるため
    - R11_G11_B10_FLOAT(符号無し)の倍の帯域

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OS側遅延OETF(scRGB出力)時の留意点

- UI描画も原則BT.709色域で描画
  - BT.2020色域で描画したい場合はややペナルティが発生
    - 方法１：UI描画の出力をBT.709色域に変換する
      - UI描画で毎回変換が必要
      - 負の数が発生する可能性の考慮
        - » 条件によってはブレンドが正しくできない
        - » SDR時はBT.709に色域制限すべき
    - 方法２：トーンマップ時にBT.2020化し、最後にBT.709に戻す
      - OS層によるOETF変換の前に、それをキャンセルするためのパスが必要
  - →scRGB出力ではBT.2020色域でのUI描画は推奨しない

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OS側遅延OETF(scRGB出力)時の留意点

- scRGB出力の1.0は80nt相当
  - 常に80nt換算でscRGBに出力する必要がある
    - つまり、80ntを1.0に、10,000ntを125.0にスケール
  - トーンマップ／OETFの白レベル(前述)とは別に考慮
    - 前述の方は便宜上コンテンツ白レベルとします

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# コンテンツ白レベルとscRGBの白レベル

- コンテンツ白レベル
  - SDRの白が現実にはどの程度の明るさとみなすか
  - 任意に調整できる
    - 一般的には100～400nt程度
    - HDRを活用するためには出来るだけ低い方が理想(200以下を推奨)
- scRGB白レベル
  - OS上のSDR基準輝度(=デスクトップの明るさとも言える)
  - OS／ドライバ層によるPQ OETF変換の基準
    - 原則として80nt固定
- これらを区別してそれぞれ正しく適用する必要がある

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# コンテンツ白レベルによる飽和輝度計算(再掲)

- トーンマップ時の飽和輝度(相対輝度)の計算
  - $n = m/w_c$
    - $n$ : 相対飽和輝度(1.0 = 白)
    - $m$ : ディスプレイ最大絶対輝度(nt)
    - $w_c$: コンテンツ白レベル(nt)
- 計算例
  - 条件
    - 最大出力10,000nt
    - コンテンツ白レベル200nt
  - 飽和輝度 = 10,000nt / 200nt = 50.0
    - 50.0が10,000ntを意味する

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# コンテンツ白レベルによる飽和輝度計算(計算例のつづき)

- 計算例
  - 条件
    - 最大出力10,000nt
    - コンテンツ白レベル200nt
  - 飽和輝度 = 10,000nt / 200nt = 50.0
    - 50.0が10,000ntを意味する
    - 最大出力を10,000ntとした場合トーンマップも50.0で飽和
    - 最大1,000ntとした場合トーンマップは1,000nt / 200nt = 5.0で飽和

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# コンテンツ白レベルによるPQ OETF(再掲)

- PQ OETFの入力は絶対輝度(nt: cd/m²)
  - 正確には1.0を10,000ntとみなす正規化値(= nt値 / 10,000)
- PQ OETFのための絶対輝度を求める
  - 露出済の相対輝度(レンダリング結果)から絶対輝度(nt)へ換算
    - 露出済レンダリング結果(相対輝度)は1.0が白なので
    - $L = w_c x$
      $L$ : 絶対輝度(nt)
      $x$ : トーンマップ／露出済相対輝度(1.0 = 白)
      $w_c$ : コンテンツ白レベル(nt)
- 換算後の絶対輝度$l$を正規化してPQ OETFを適用
  - $y = {}^{L}/_{10000} = {}^{w_c x}/_{10000}$
  - $V = \mathrm{PQ_OETF}(y)$
- ただしscRGBでは原則PQ OETFは適用しない
  - LUTやディザのために一時色空間として必要になるケースはある(後述)

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# コンテンツ白レベルと scRGB白レベルによる調整スケール

- トーンマップ後相対輝度から絶対輝度およびscRGB出力値への変換
  - $L = w_c x$
    $L$ : 絶対輝度(nt)
    $x$ : 露出／トーンマップ済相対輝度(1.0 = 白)
    $w_c$ : コンテンツ白レベル(nt)
  - $s = {}^{L}/_{w_s} = {}^{w_c x}/_{w_s}$
    $s$ : scRGB出力値(= 絶対輝度nt / scRGB白レベル)
    $w_s$: scRGB白レベル(nt) = 80

- 計算例
  - コンテンツ白レベル$w_c = 200$, scRGB白レベル$w_s = 80$ の例
    - $x = 0.4$(80nt: scRGB白相当) → $s = 1.0$
    - $x = 1.0$(200nt: コンテンツ白) → $s = 2.5$
    - $x = 50$(10,000nt: PQ最大輝度) → $s = 125$

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OS側遅延OETF(scRGB出力)時の留意点

- scRGB出力時のディザリング
  - OS／ドライバ層ではサポートされない
  - ディザはOETF後の非線形出力色空間で適用する必要あり
    - 最終的な色信号上の分解能で適用する必要があるため
  - →アプリ側で一度出力色空間に変換して適用
    - 出力OETF色空間に変換→ディザリング→scRGBに戻す
    - やや無駄が多いが、最後にパスを追加するよりは効率的

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# その他の考慮すべきこと

本セッションの目的
色空間
前提として押さえておきたいこと
基本的なレンダリングフローの検討
その他の考慮すべきこと
互換性重視レンダリングフローの構築
まとめ

Silicon Studio ©Silicon Studio Corp. all rights reserved!

# フローの詳細な考慮

- 先述のフローに含まれていない細かな事項の検討
  - 動的トーンマップ最適化
  - LUTの互換性
  - カラーグレーディングとしてのガンマ調整
- 詳細を考慮した色空間フロー


©Silicon Studio Corp., all rights reserved.

# フローの詳細な考慮

- 先述のフローに含まれていない細かな事項の検討
  - 動的トーンマップ最適化
  - LUTの互換性
  - カラーグレーディングとしてのガンマ調整
- 詳細を考慮した色空間フロー

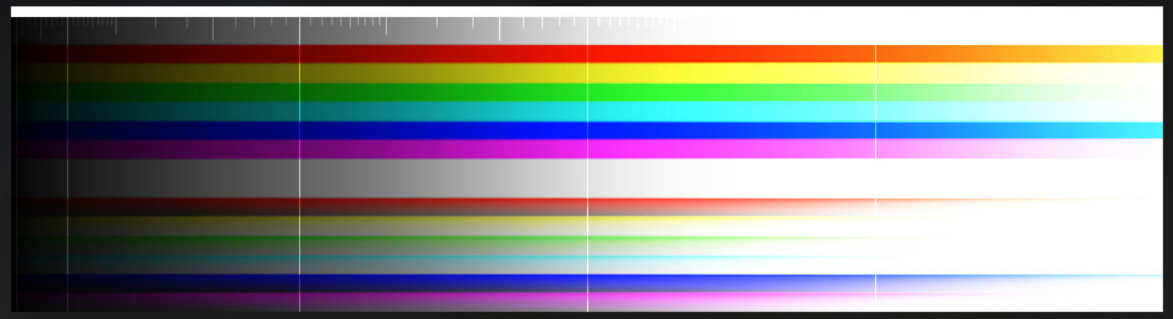

©Silicon Studio Corp., all rights reserved.

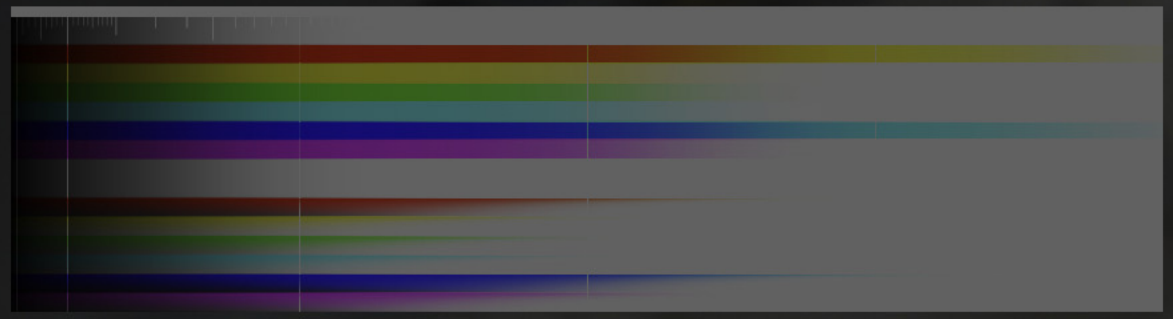
# シーン内の最大輝度に応じて動的にトーンマッピングを最適化

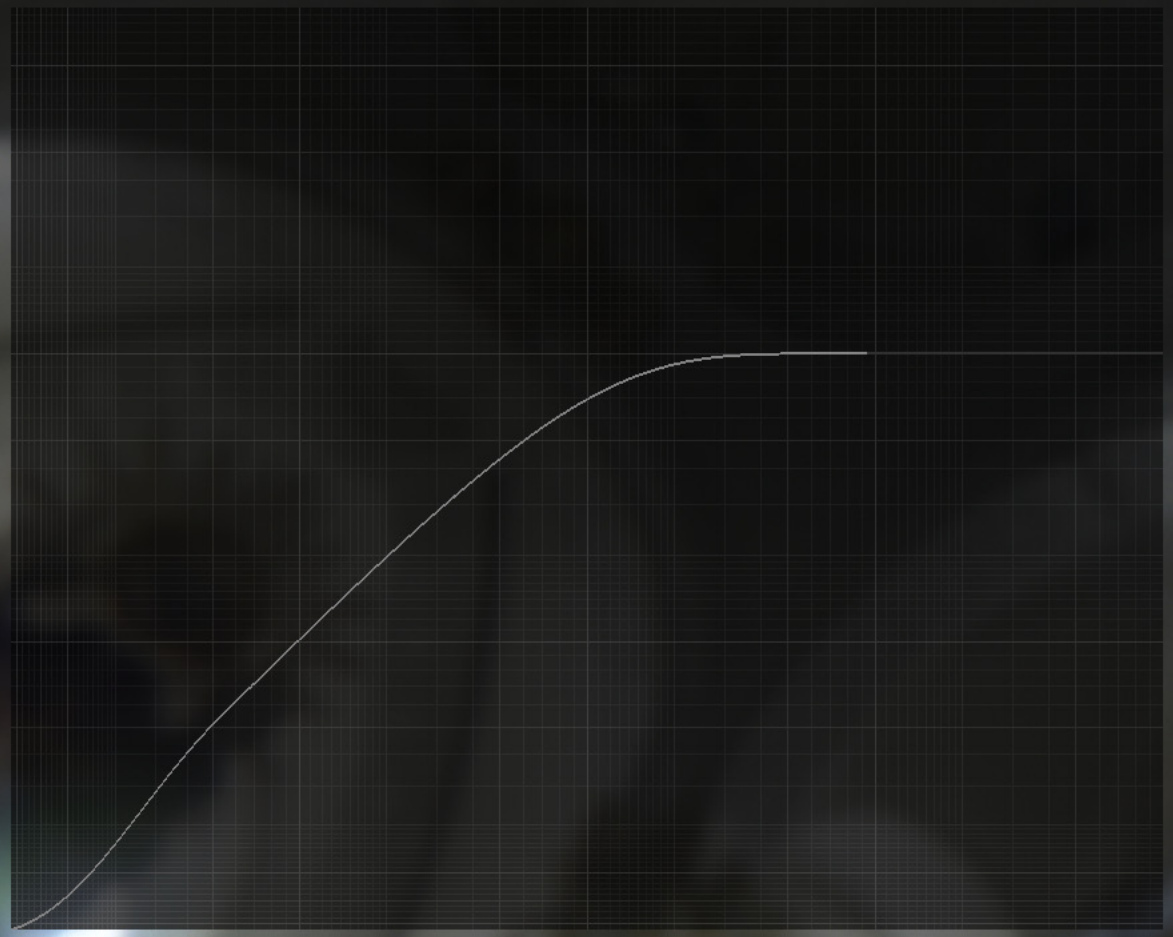
- 最大出力輝度よりもシーンが明るい場合
  - 高輝度領域を圧縮する
    - シーン内最大輝度を最大出力値にマッピング
- 最大出力輝度よりもシーンが暗い場合
  - 高輝度領域も圧縮せず線形に

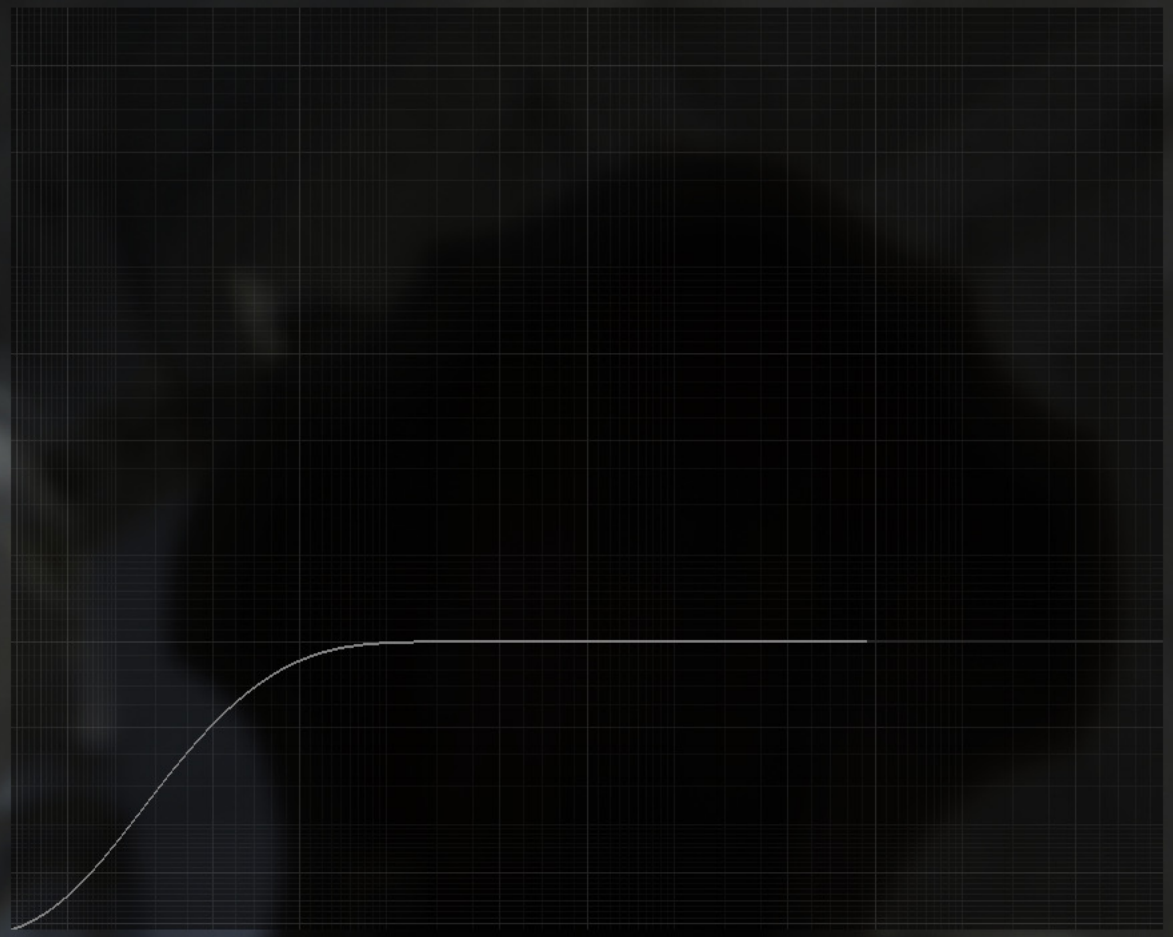
- 以下のサンプル画像のトーンマップは1,000nt飽和
  - 1,000ntの信号まで有効に階調表示できるテレビ用の設定


©Silicon Studio Corp., all rights reserved.

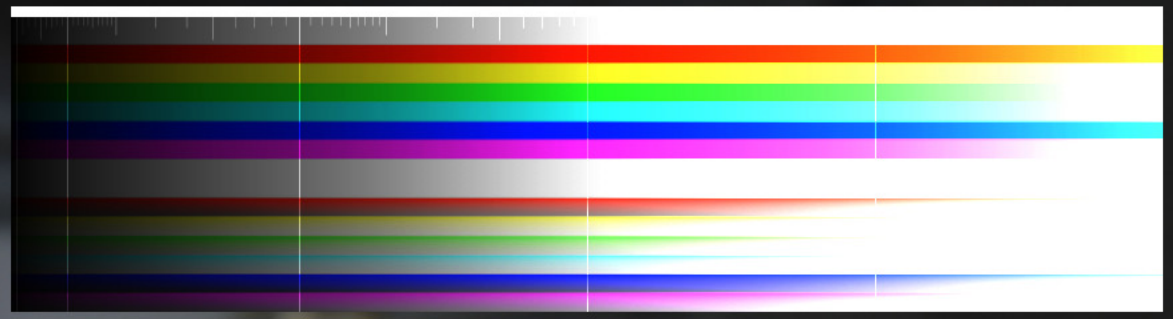


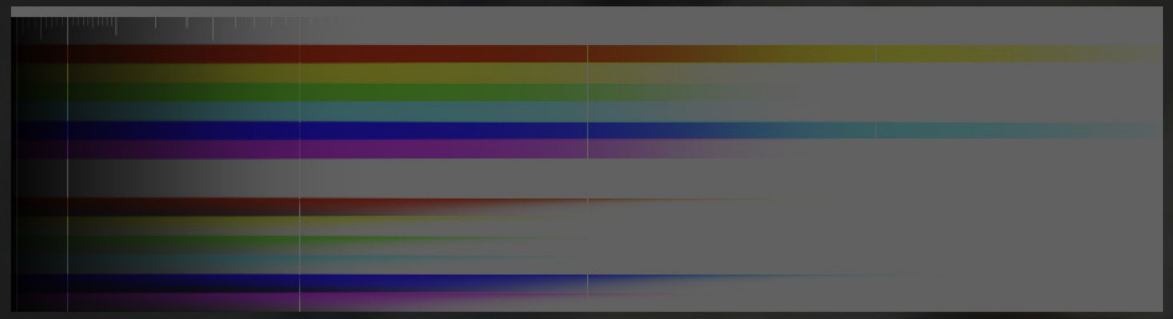


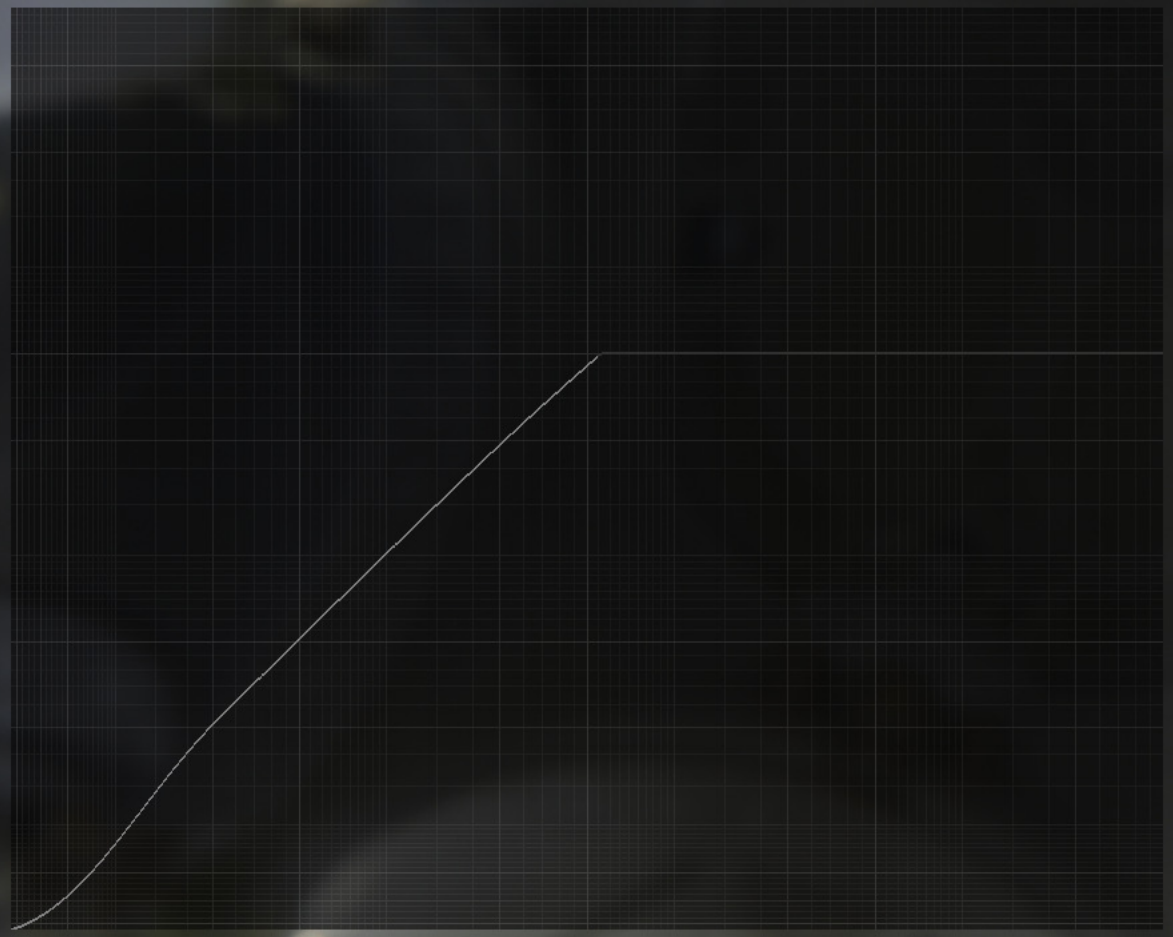


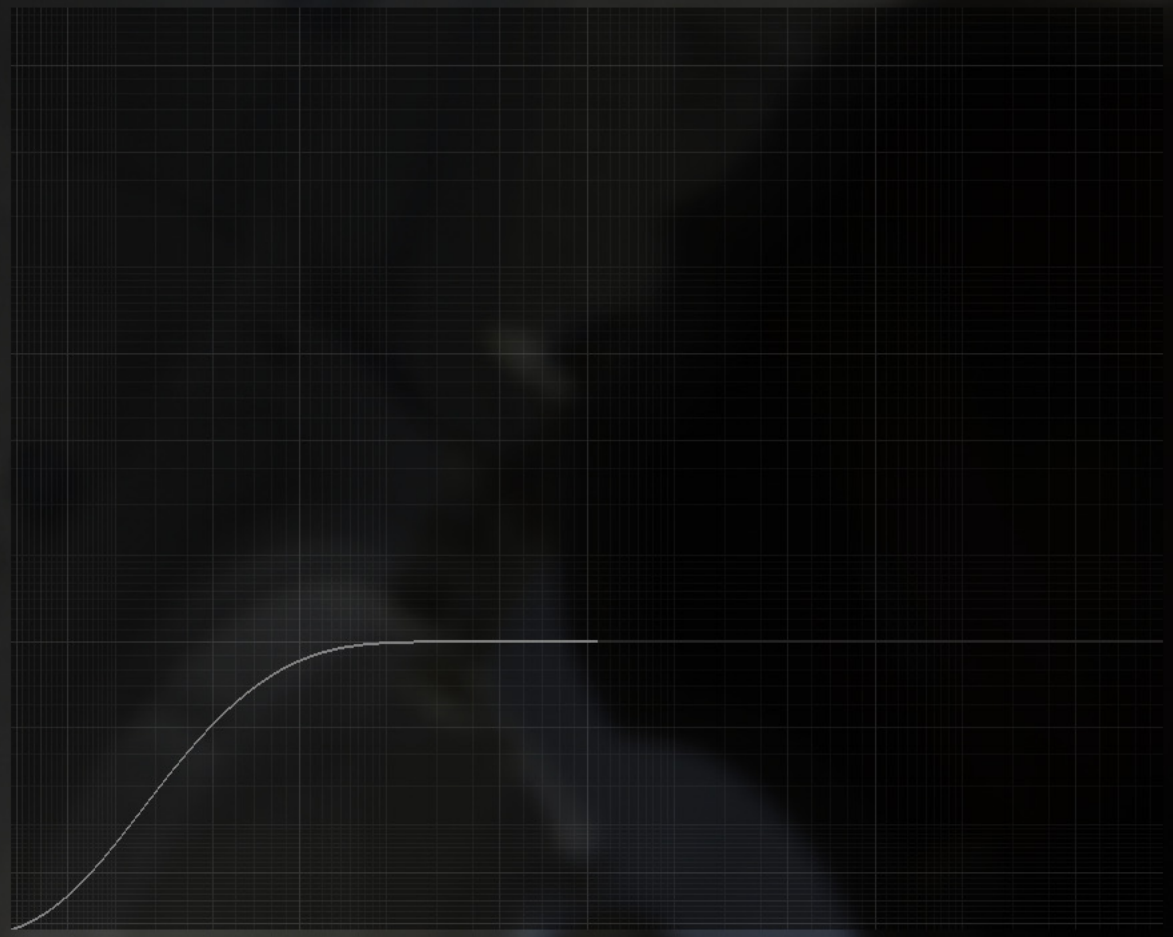


動的トーンマップ(高輝度ピクセルをもつシーン)

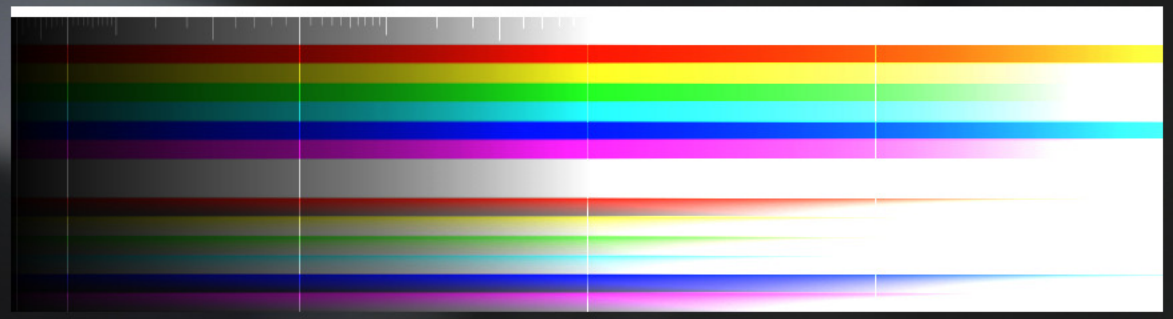


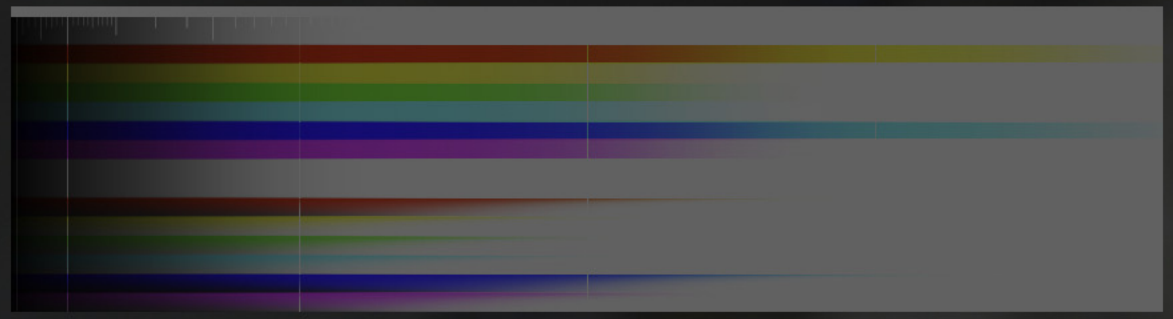


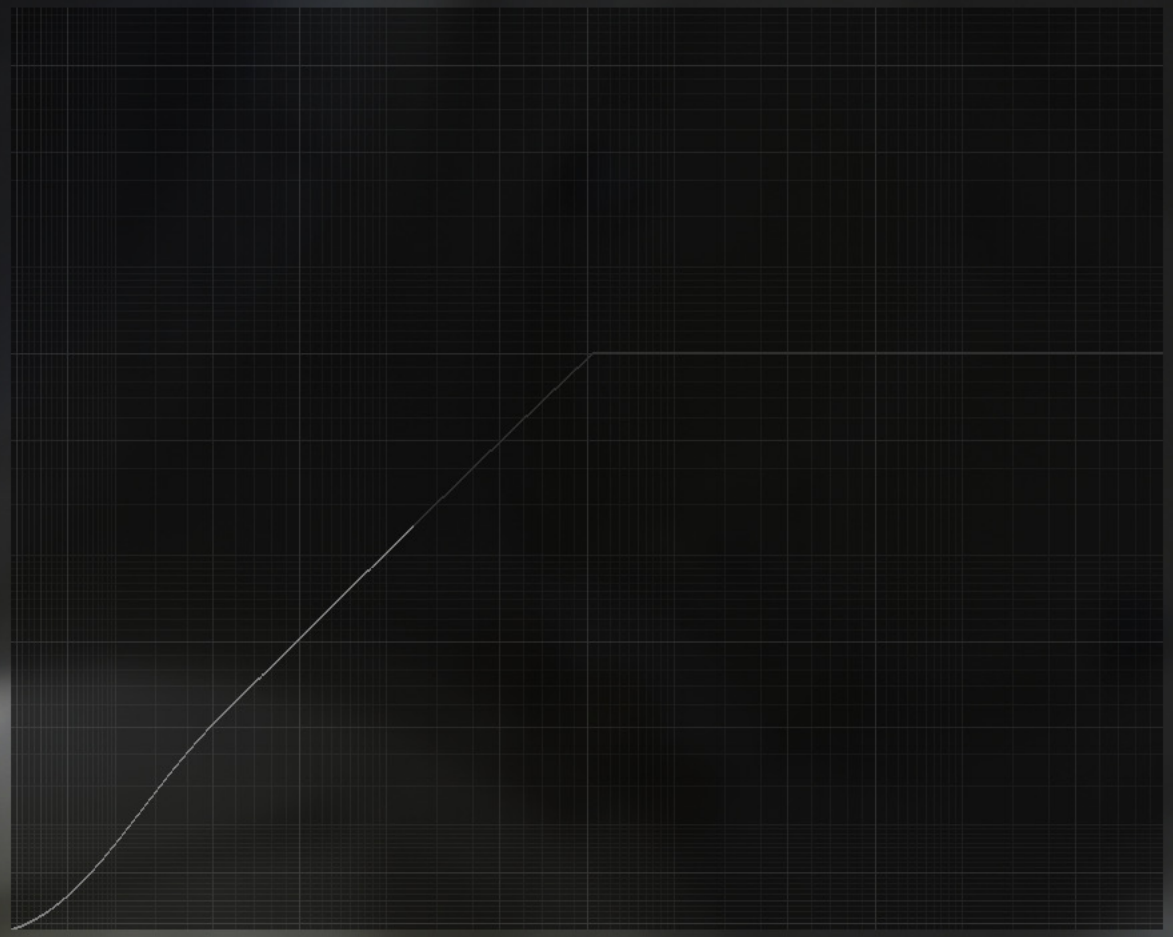


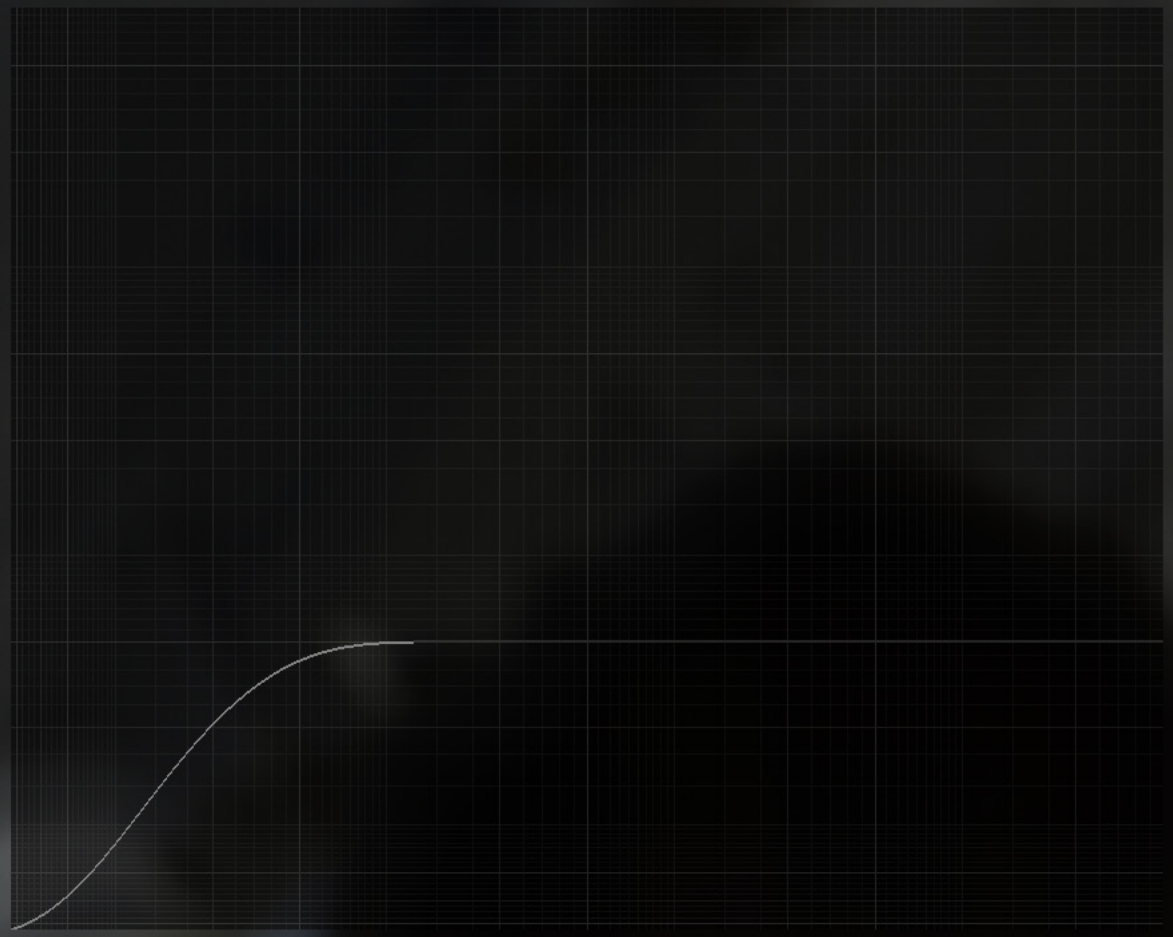


動的トーンマップ(中輝度のピクセルをもつシーン)







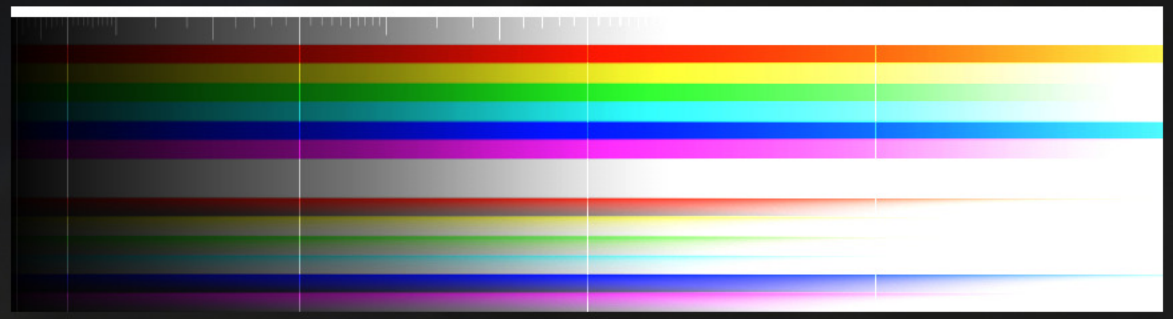


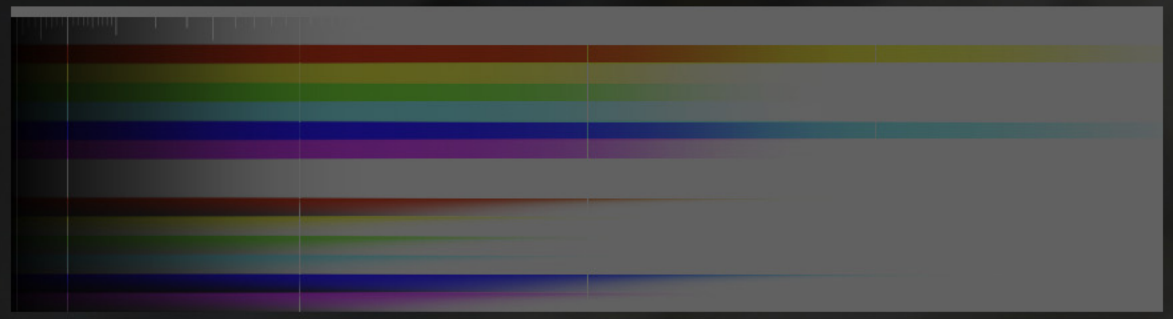
動的トーンマップ(低輝度ピクセルのみのシーン)

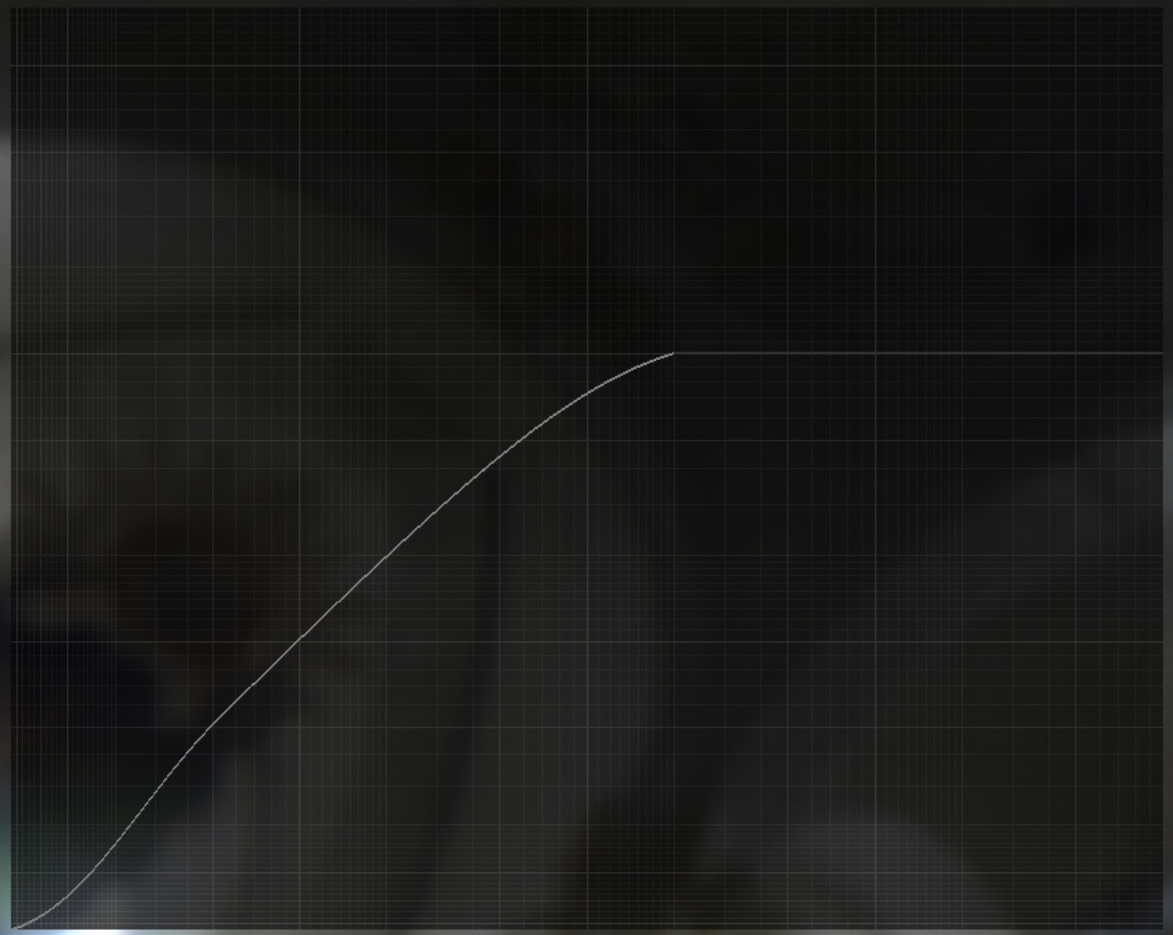
# シーン輝度が2,000ntと仮定したマッピング

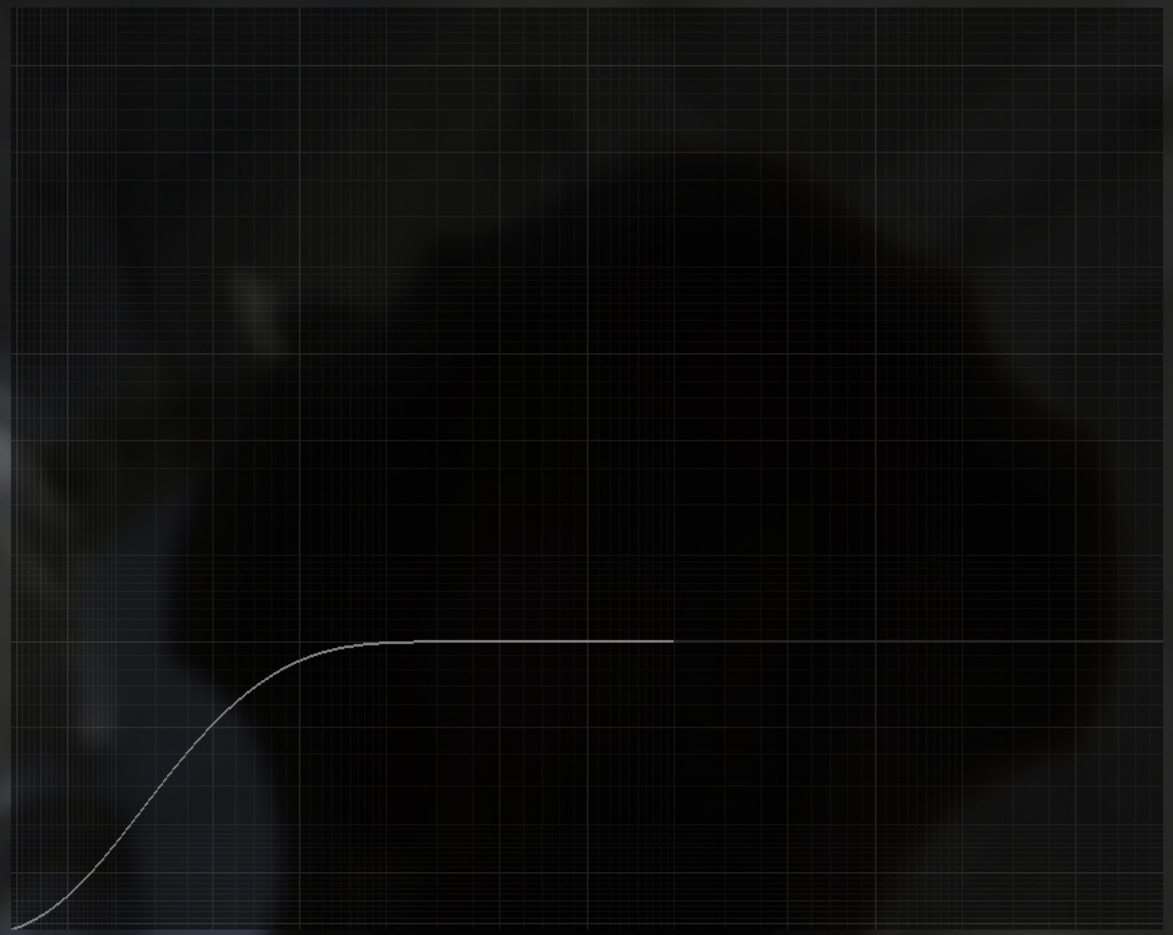
- シーンの最大輝度が常に2,000nt固定と仮定
  - 常に2,000nt時の出力を1,000ntにマッピング
  - 2,000ntよりもシーンが明るい場合
    - 2,000ntまでしか階調を表現できず白飛び
  - 2,000ntよりもシーンが暗い場合
    - 1,000ntよりも暗く表示され、テレビの能力を活かせない

Silicon Studio ©Silicon Studio Corp., all rights reserved.

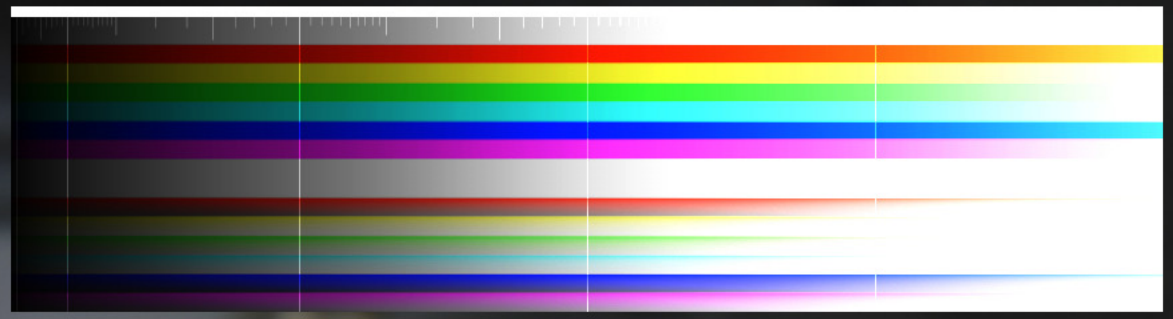


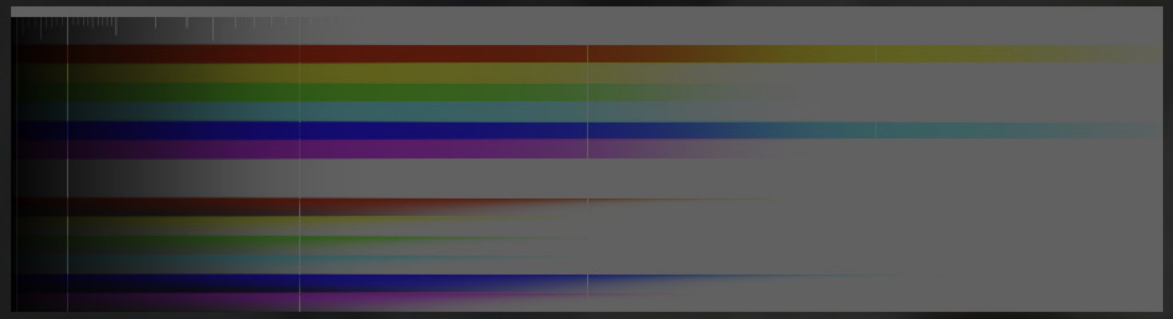


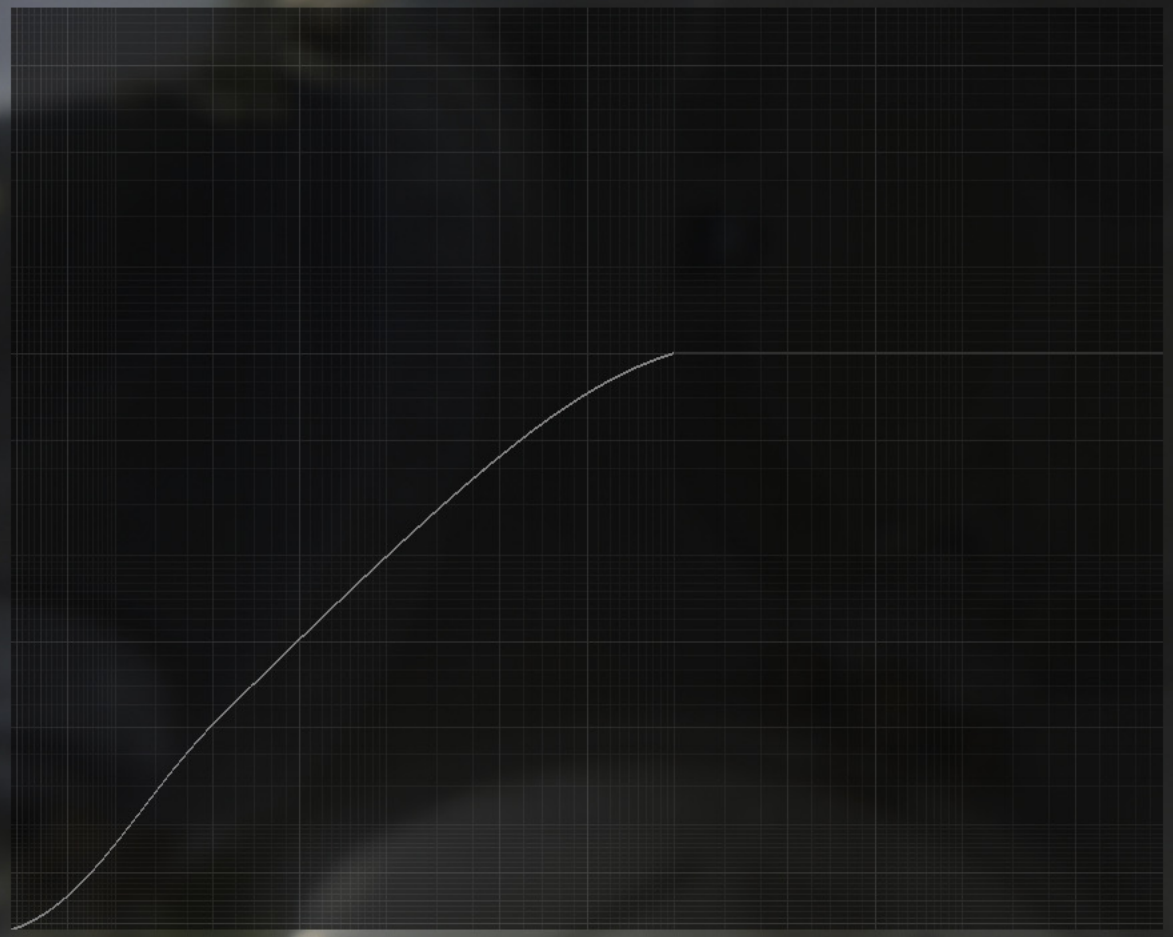


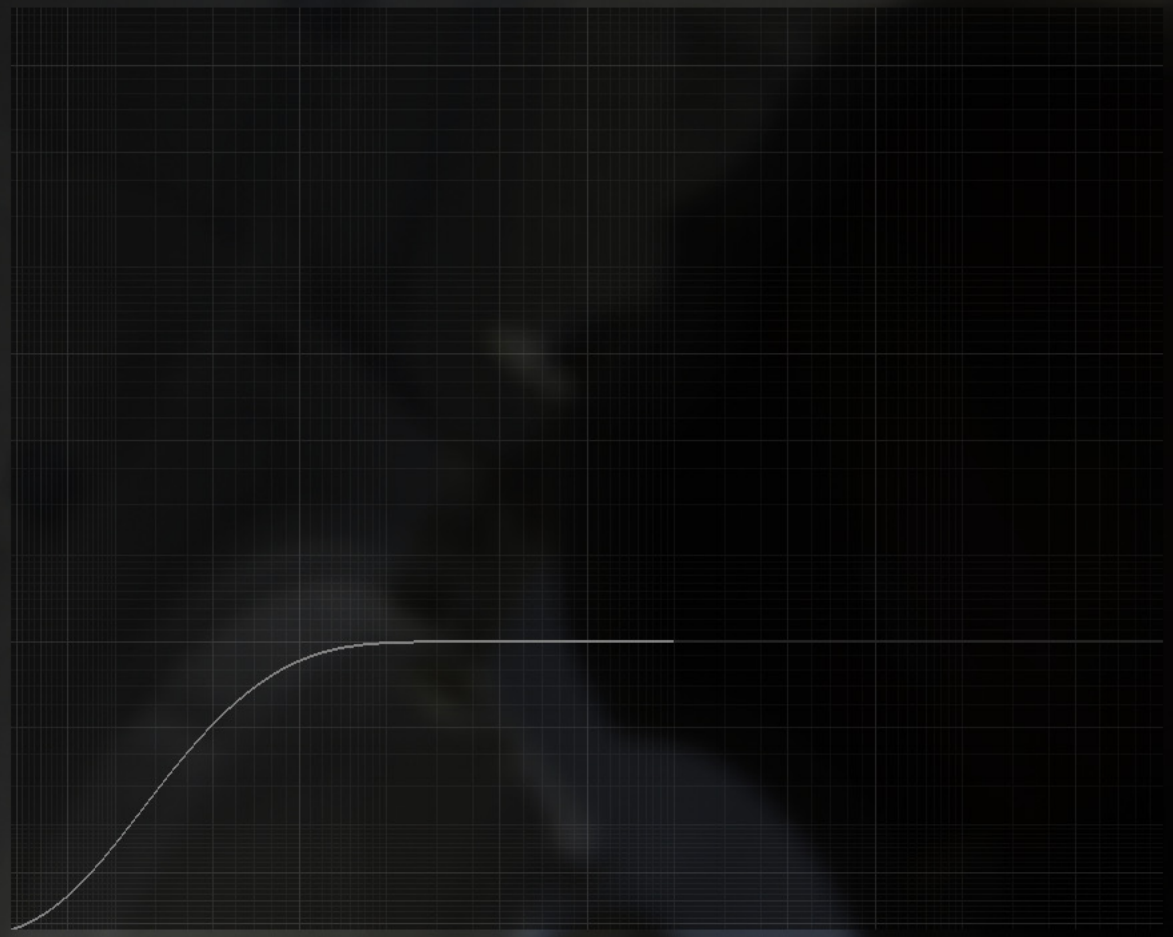


2,000ntシーン用静的トーンマップ(高輝度ピクセルをもつシーン)







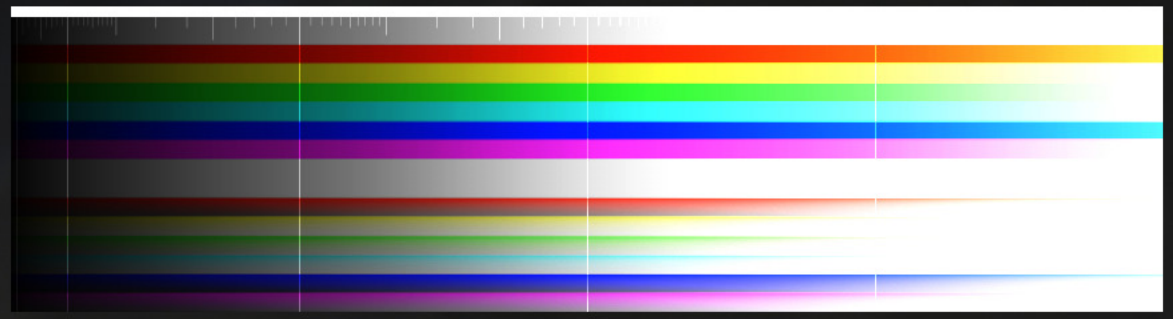


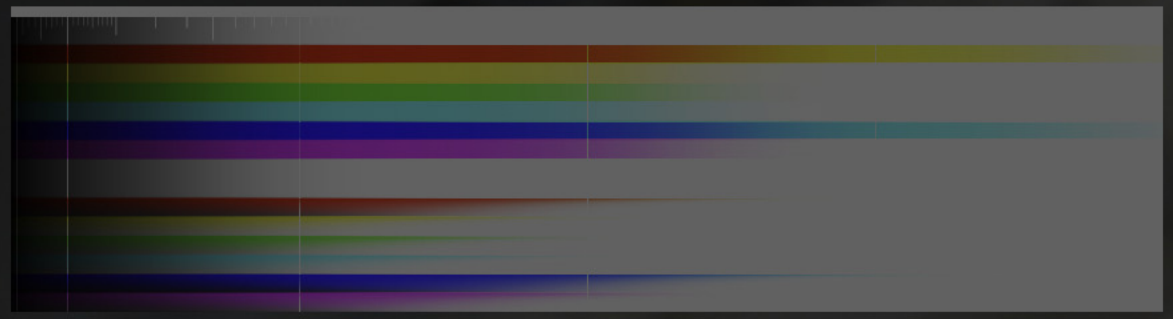
2,000ntシーン用静的トーンマップ(中輝度のピクセルをもつシーン)

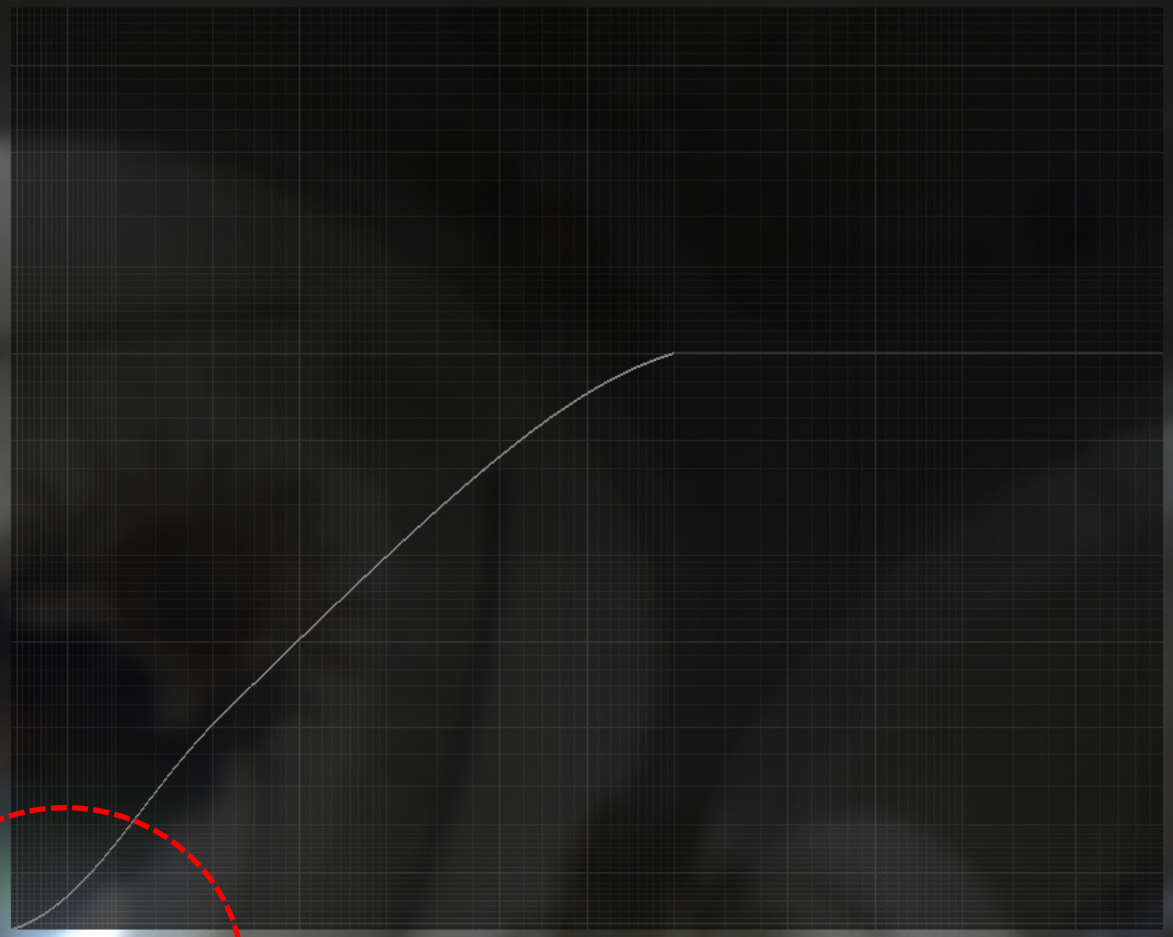
# 2,000nt固定と動的マッピングの比較

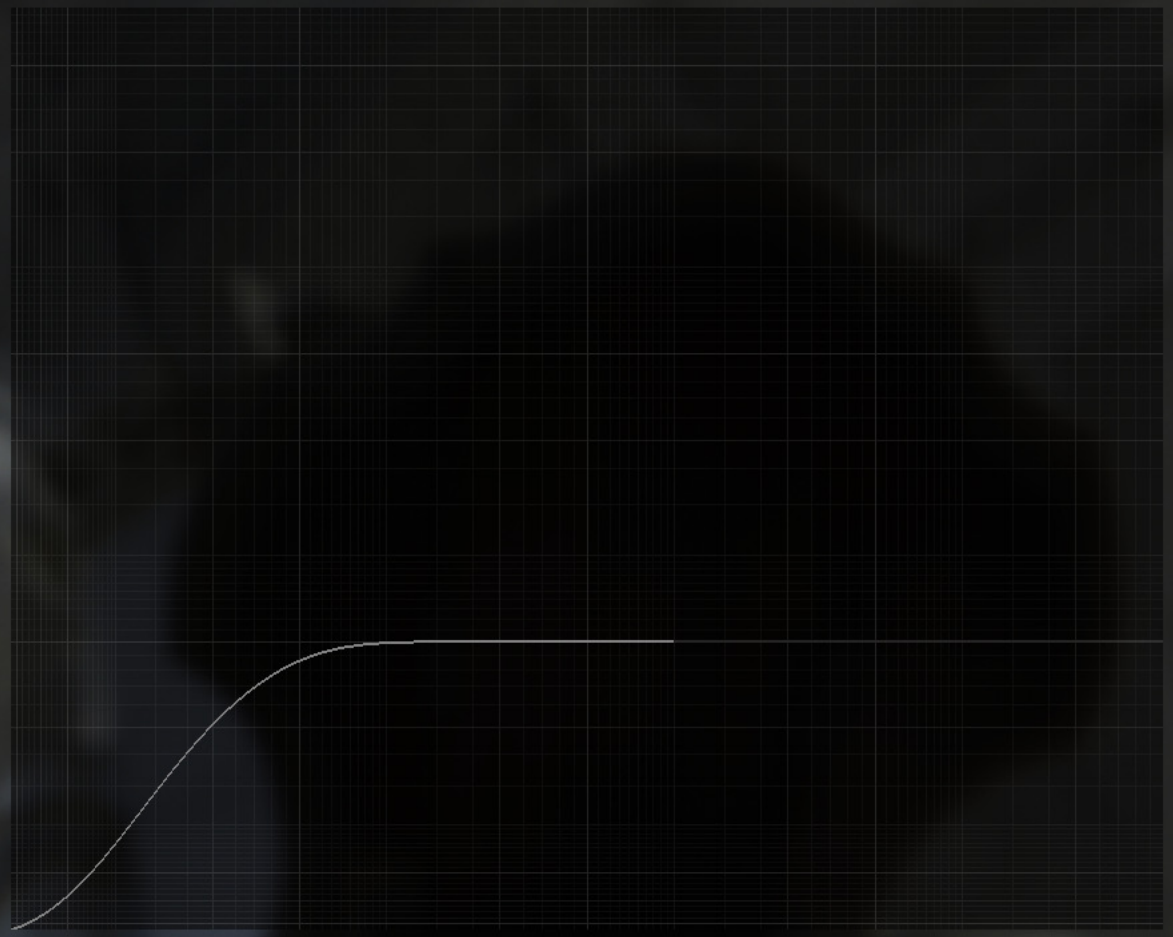
- シーンの最大輝度が常に2,000nt固定と仮定
  - 常に2,000nt時の出力を1,000ntにマッピング
- シーンの最大輝度によって動的にマッピング
  - トーンマップの最大出力値1,000ntを最大限に利用

Silicon Studio ©Silicon Studio Corp., all rights reserved.

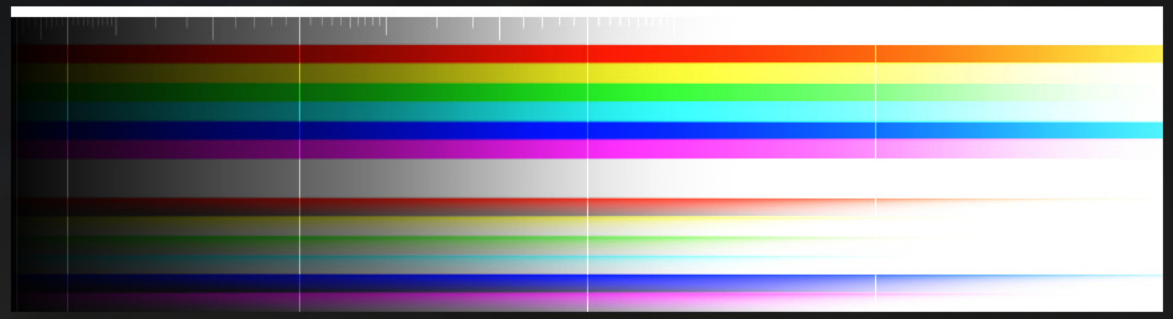


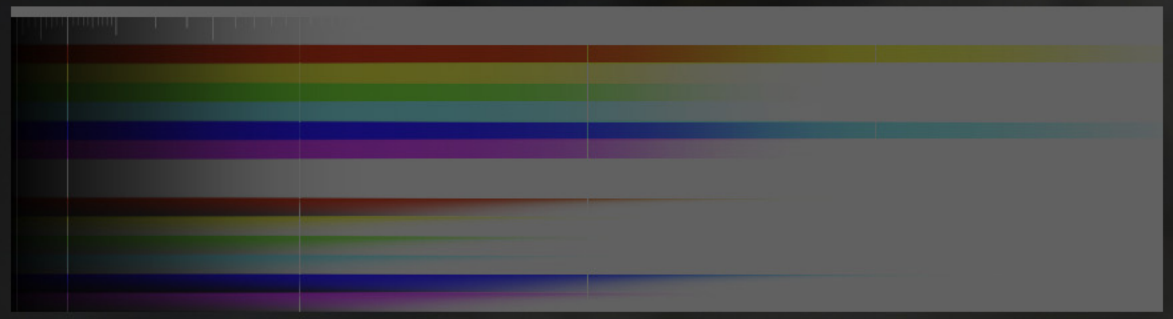


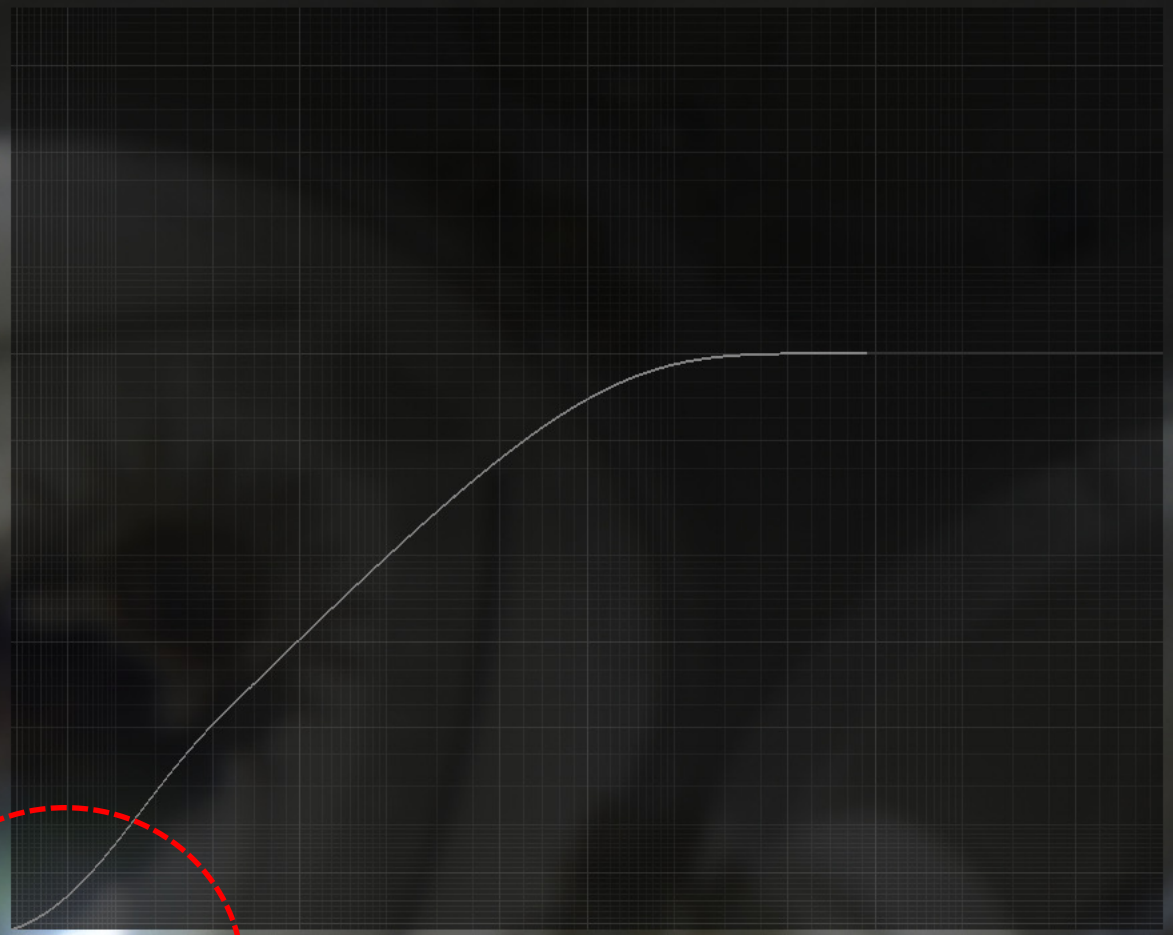


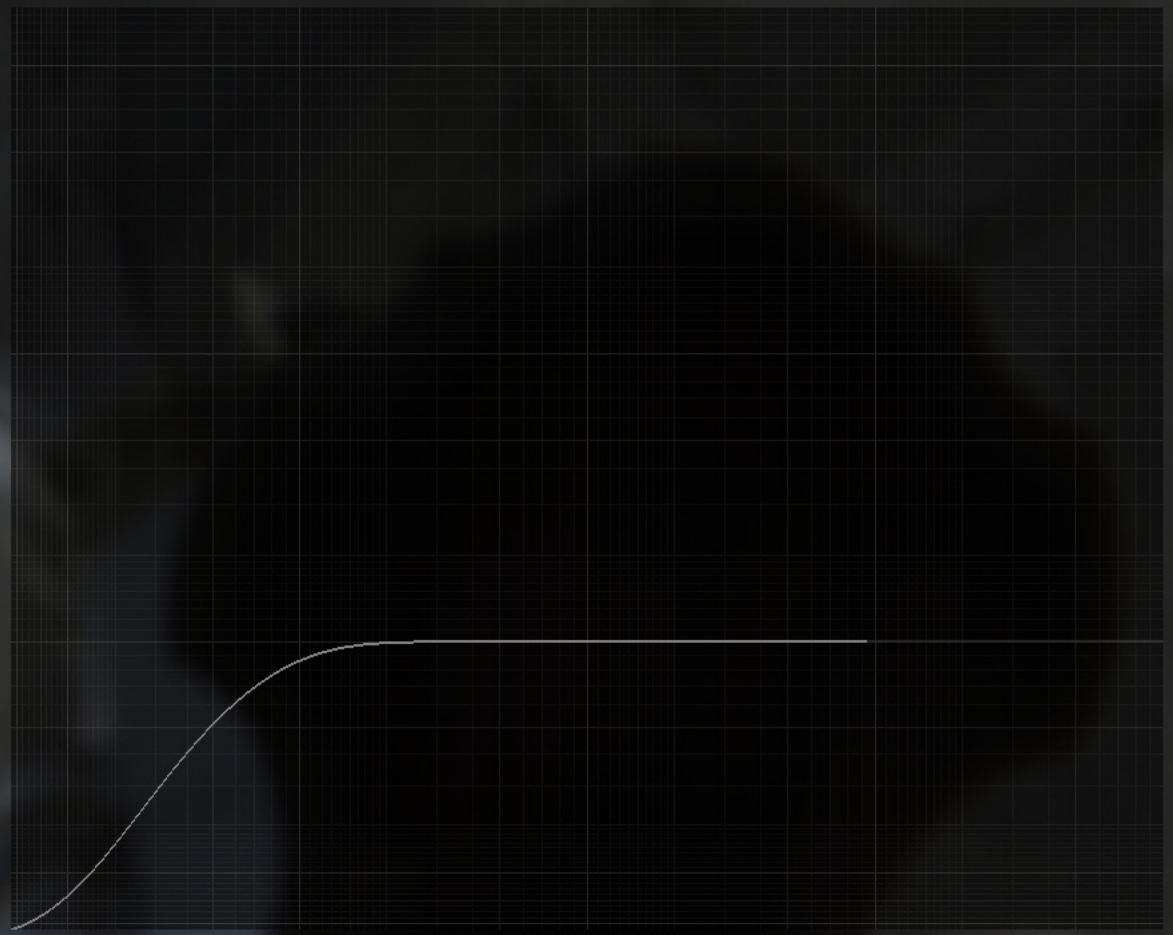


2,000ntシーン用静的トーンマップ(高輝度ピクセルをもつシーン)

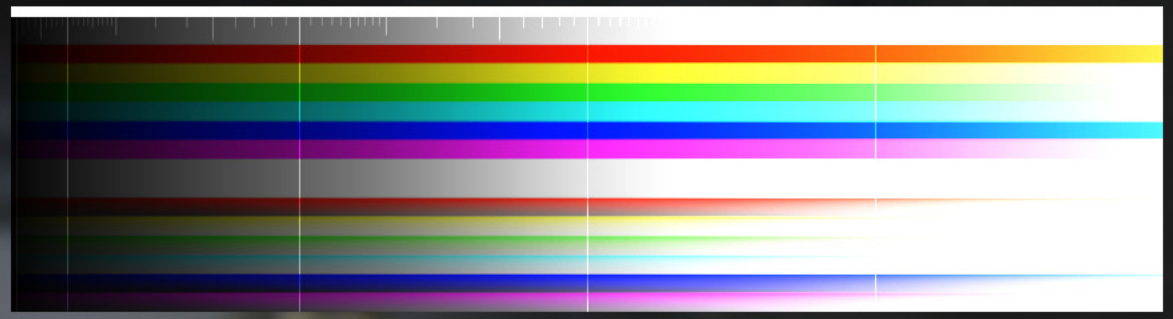


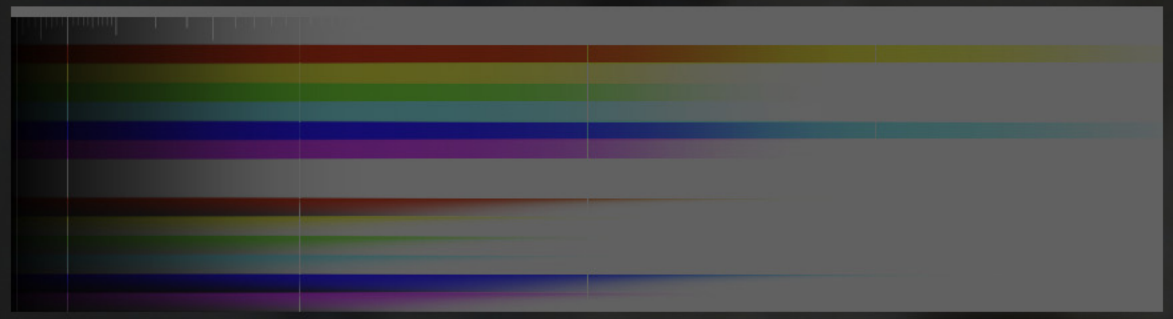


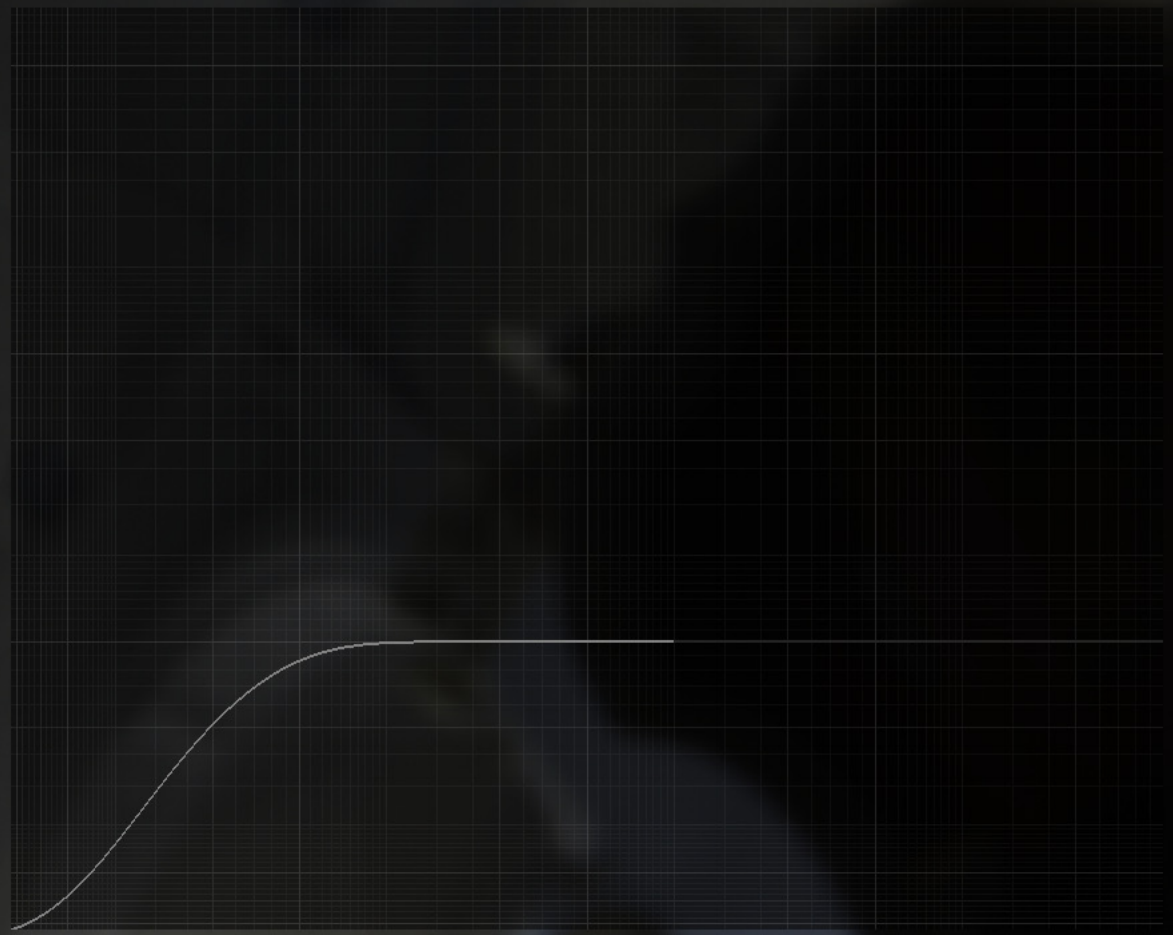




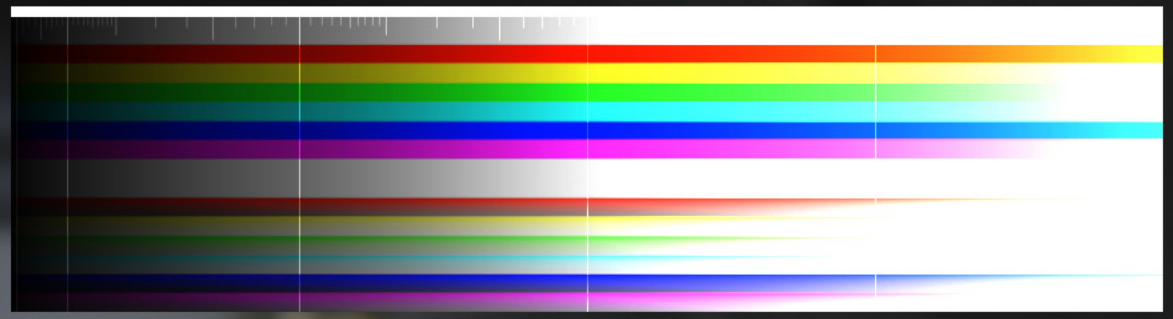
動的トーンマップ(高輝度ピクセルをもつシーン)

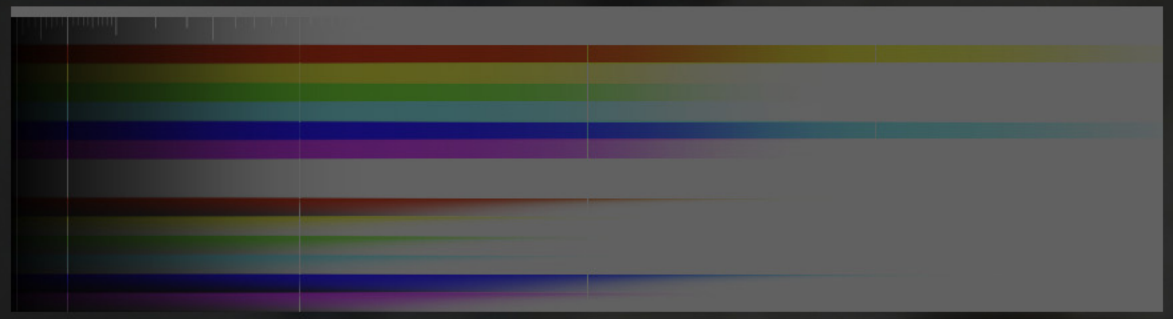


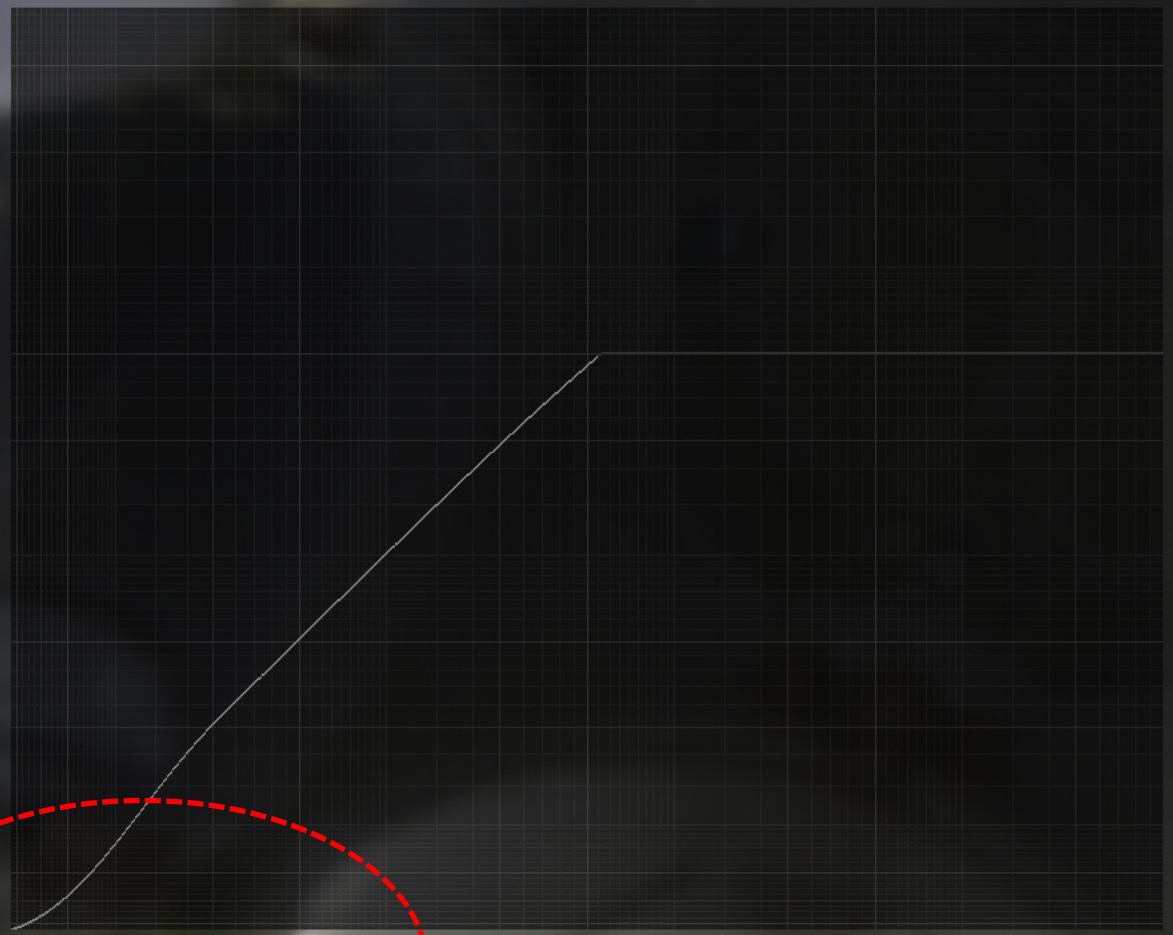


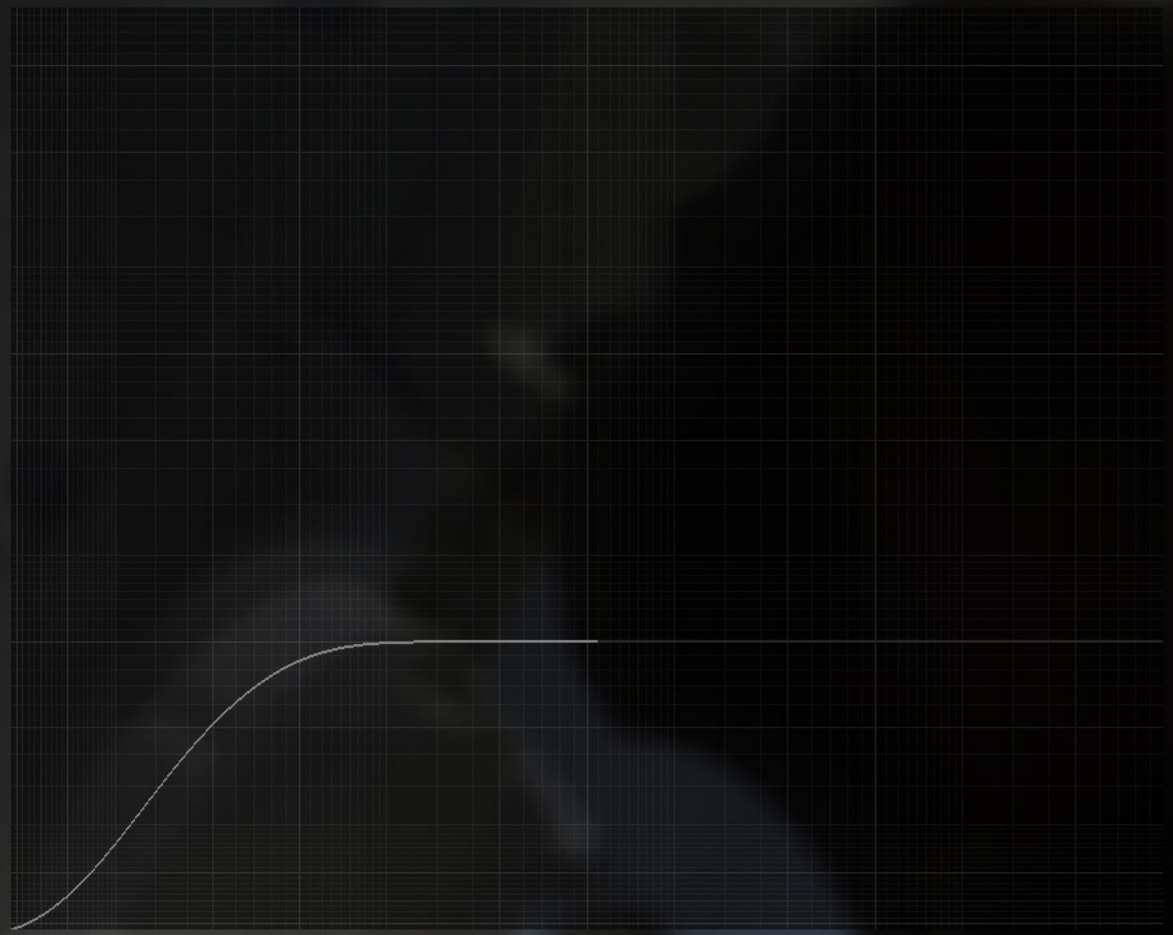


2,000ntシーン用静的トーンマップ(中輝度のピクセルをもつシーン)









動的トーンマップ(中輝度のピクセルをもつシーン)

# 動的メタデータへの対応

- 解析した最大輝度は動的メタデータに応用できる
  - Dolby Vision
  - HDR10+(HDR10 Plus)を含む今後の拡張

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 動的メタデータへの対応

- 解析されたシーン最大輝度をトーンマップ
  - 解析はトーンマップ前の線形HDR色空間で行っている
    - 動的トーンマップ時に必要な情報のため
  - メタデータには出力信号の最大輝度が必要
    - よって最大輝度もトーンマップ後の値が必要


©Silicon Studio Corp., all rights reserved.

# 動的メタデータへの対応

- 一般的には絶対輝度(nt値)で指定する必要がある
  - トーンマップ済みシーン最大輝度を
    白レベルで乗算して絶対輝度(nt)に換算(式は既出)
    - $L = w_c x$

      $L$ : メタデータ用シーン最大絶対輝度(nt)

      $x$ : 露出／トーンマップ済シーン最大輝度(1.0 = 白)

      $w_c$ : コンテンツ白レベル(nt)
  - 計算例
    - トーンマップ済みシーン最大輝度(相対輝度)が10.0、
      コンテンツ白レベルが100ntの場合
      - メタデータ用シーン最大絶対輝度 = 10.0 x 100nt = 1,000nt

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 動的メタデータも必要？

- 動的トーンマップしたのにメタデータも必要？
  - 実際にどのように表示されるかを考えてみる
  - 例えば最大500ntの明るさで
    2,000ntの信号まで有効に階調表示できるテレビと仮定
    - 1,000nt信号が400nt程度に表示される
    - 2,000nt信号が500ntで表示される
  - 実際にはテレビやメーカ毎に異なる
    - ここで挙げるのはあくまで考察用に仮定したもの


©Silicon Studio Corp., all rights reserved.

# 最大輝度500ntの明るさで 2,000ntの信号まで階調表示可能なテレビ

- アプリ側動的トーンマップ最適化により
  - 線形シーン最大輝度が500ntの場合最大500ntの信号を生成
  - 線形シーン最大輝度が1,000ntの場合最大1,000ntの信号を生成
- 動的メタデータが存在すると
  - ○ 最大500ntのシーン(500nt信号出力)は500ntの輝度で適切に表示される
  - ○ 最大1,000ntのシーン(1,000nt信号出力)は500ntの輝度で適切に階調表示される
- 動的メタデータが存在しないと
  - 静的メタデータ500ntの場合:
    - ○ 最大500ntのシーン(500nt信号出力)は500ntの輝度で適切に表示される
    - × 最大1,000ntのシーン(1,000nt信号出力)は500nt以上が白飛び
  - 静的メタデータ2,000ntの場合:
    - △ 最大500ntのシーン(500nt信号出力)は400nt程度に表示される
    - △ 最大1,000ntのシーン(1,000nt信号出力)は450nt程度に表示される


©Silicon Studio Corp., all rights reserved.

# 最大輝度500ntの明るさで2,000ntの信号まで階調表示可能なテレビ

- アプリ側では最大2,000nt出力の静的トーンマップ(最適化無効)の場合
  - 線形シーン最大輝度が500ntの場合最大400nt程度の信号を生成
  - 線形シーン最大輝度が1,000ntの場合最大700nt程度の信号を生成
- 動的メタデータが存在すると
  - △ 最大500ntのシーン(400nt信号出力)は400ntの程度の輝度で表示される
  - ○ 最大1,000ntのシーン(700nt信号出力)は500ntの輝度で適切に階調表示される
- 動的メタデータが存在しないと
  - 静的メタデータ500ntの場合:
    - △ 最大500ntのシーン(400nt信号出力)は400ntの輝度で表示される
    - △ 最大1,000ntのシーン(700nt信号出力)は600nt強以上が白飛び
  - 静的メタデータ2,000の場合:
    - × 最大500ntのシーン(400nt信号出力)は400ntよりも暗めに表示される
    - × 最大1,000ntのシーン(700nt信号出力)は400nt程度に表示


©Silicon Studio Corp., all rights reserved.

# 動的メタデータも必要？

- 最適な表示のためにはどちらも有益
  - どちらか片方だけでは最適な結果を得られない
- 動的トーンマップが前提として必要
  - その上で動的メタデータを利用できるなら活用すべき


©Silicon Studio Corp., all rights reserved.

# 動的トーンマップ／メタデータ留意点

- 画面輝度解析のため少しオーバヘッドが生じる
  - コストパフォーマンスで判断
- フリッカリング
  - 解析結果を直接使うとフリッカリングが生じやすい
  - →フレーム間で解析結果を補間して徐々に適応


©Silicon Studio Corp., all rights reserved.

# フローの詳細な考慮

- 先述のフローに含まれていない細かな事項の検討
  - 動的トーンマップ最適化
  - LUTの互換性
  - カラーグレーディングとしてのガンマ調整
- 詳細を考慮した色空間フロー


©Silicon Studio Corp., all rights reserved.

# LUTの互換性

- カラーグレーディングにLUTを使用する方法
- 一般にsRGBなどの出力色空間で適用されることが多い
  - SDR: BT.709 sRGB
  - HDR: BT.2020 PQ
- SDRとHDRの互換性を保てない

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# LUTの互換性

- 新規タイトル開発の場合
  - 可能ならPQのようなHDR色空間で作成した方が良い
    - OETF後ではなくトーンマップ前の線形HDRの段階で適用
    - HDRで制作したLUTはSDR出力にもそのまま適用できる
  - HDR LUTについての参照
    - “Advanced Techniques and Optimization of HDR VDR Color Pipelines” GDC2016 [Lottes16]
    - 精度や演算効率について詳細に検証されており大変参考になる


©Silicon Studio Corp., all rights reserved.

# LUTの互換性

- 既存タイトルのHDR拡張対応の場合
  - 既存のLUTを利用するにはsRGB色空間で適用する必要あり
  - HDR拡張トーンマップの前にSDR LUTシミュレーション
    - BT.709レンダリング結果→SDRトーンマップ→sRGB OETF
      - sRGB色空間でLUT適用
    - sRGB EOTF→逆SDRトーンマップ→BT.2020色域非線形拡張変換
  - SDRシステムガンマ(OOTF)互換カーブと同時適用が効率的
    - OOTF互換カーブもSDRトーンマップ色空間で行う
    - 上記のsRGB EOTFをBT.1886(γ2.4) EOTF (PCではγ2.2 EOTF)に変更

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# sRGB LUTと互換性を保つ場合の注意

- 負荷が高い
  - 適切な色空間で処理するための変換フローが複雑
  - フローすべてを1つのHDR LUTに格納する事は可能
- 高輝度領域の精度が低い
  - 高輝度部分の色変化が過敏になりやすい
    - SDR LUT制作時には想定されていない変換であるため
    - 少し暗い色でも最大値に置換される場合が(割と頻繁に)ある
  - HDR色空間に戻すまでの計算精度に注意
    - 最大値を逆トーンマップした結果無限大にならないように
    - →SDRトーンマップは有限の値をマッピングするように修正すべき


©Silicon Studio Corp., all rights reserved.

# HDR出力時のLUTの注意

- 8-bit RGBでは精度不足
  - 32x32x32以上の10bit以上を推奨
    - GPU側の補間精度にも注意
      - 8-bitフォーマットでは補間も8bit精度になる場合も
    - →既存のLUTも事前に変換しておく
      - グレーディングパラメタが残っているなら直接再作成するのが理想
  - 前後の色空間変換も含めるHDR LUTなら
    - RGBA16_FLOAT
    - 補間精度さえ高ければもっと低精度のFLOATでも問題ない？
      - R11_G11_B10_FLOATは？
      - 最高輝度付近の色がやや精度不足かも知れない


©Silicon Studio Corp., all rights reserved.

# フローの詳細な考慮

- 先述のフローに含まれていない細かな事項の検討
  - 動的トーンマップ最適化
  - LUTの互換性
  - カラーグレーディングとしてのガンマ調整
- 詳細を考慮した色空間フロー


©Silicon Studio Corp., all rights reserved.

# カラーグレーディングとしてのガンマ調整

- SDRでsRGB変換と同時にガンマ調整
  - 従来のSDRレンダリングではよく利用される
  - sRGBではなく一様なガンマ値を使っていたケースも多い
  - 例：
    - sRGBの代わりにγ2.2補正を使用
    - 高速化のためγ2.0補正(sqrt)を使用
    - コントラストを上げるためにγ1.8補正を使用
    - etc.

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# カラーグレーディングとしてのガンマ調整

- HDR時にSDRとの互換性を保つためには
  - トーンマップ後OETF変換の前にガンマ調整を適用
  - SDRシステムガンマ(OOTF)互換カーブと同時適用が効率的
    - SDRシステムガンマ(再掲)
      - sRGB OETF→BT.1886(γ2.4) EOTF(PCではγ2.2)
    - sRGB OETFの代わりにSDR時のガンマ補正をそのまま適用
      - SDR時ガンマ補正→BT.1886(γ2.4) EOTF(PCではγ2.2)
  - SDR LUTのフローも合成すると、HDR拡張トーンマップ前に
    - BT.709レンダリング結果→SDRトーンマップ
    - SDR時ガンマ補正
      - SDRガンマ空間でLUT適用
    - BT.1886(γ2.4) EOTF(PCではγ2.2)
    - 逆SDRトーンマップ→BT.2020色域非線形拡張変換


©Silicon Studio Corp., all rights reserved.

# フローの詳細な考慮

- 先述のフローに含まれていない細かな事項の検討
  - 動的トーンマップ最適化
  - LUTの互換性
  - カラーグレーディングとしてのガンマ調整
- 詳細を考慮した色空間フロー


©Silicon Studio Corp., all rights reserved.

# 細かな処理の互換性を考慮したHDR出力の色空間フロー

- 既存のSDR出力との互換性を重視する場合
  - SDRトーンマップ
    - ガンマ調整(あるいはsRGB OETF)
    - SDR LUT
    - BT.1886(PCではγ2.2) EOTF
  - SDR逆トーンマップ
  - BT.2020色域への非線形拡張変換
  - 最大輝度解析
  - HDR拡張トーンマップ
  - PQ OETF
  - ディザリング

- このフローを先述のレンダリングフローに矛盾無く組み込む

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# SDR出力の色空間フロー

- SDR出力時は従来同様の一般的なフロー
  - SDRトーンマップ
  - ガンマ調整(あるいはsRGB OETF)
  - SDR LUT
  - ディザリング
  - LUTやディザリングを利用する場合
    - 原則としてハードウェアsRGB出力は利用できない
    - sRGB変換後に各処理が必要になるため

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 互換性重視レンダリングフローの構築

本セッションの目的
色空間
前提として押さえておきたいこと
基本的なレンダリングフローの検討
その他の考慮すべきこと
互換性重視レンダリングフローの構築
まとめ

Silicon Studio
©Silicon Studio Corp., all rights reserved!

# 既存のSDR出力との互換性を重視

- 既存タイトルのHDR対応等
  - レンダリングはBT.709ベースで
    - UI描画も原則としてBT.709の素材を使用
  - LUTはsRGB(あるいは任意のガンマ補正済)色空間で適用
- さまざまなHDR方式への対応
  - 新たな方式が追加されても対応可能に

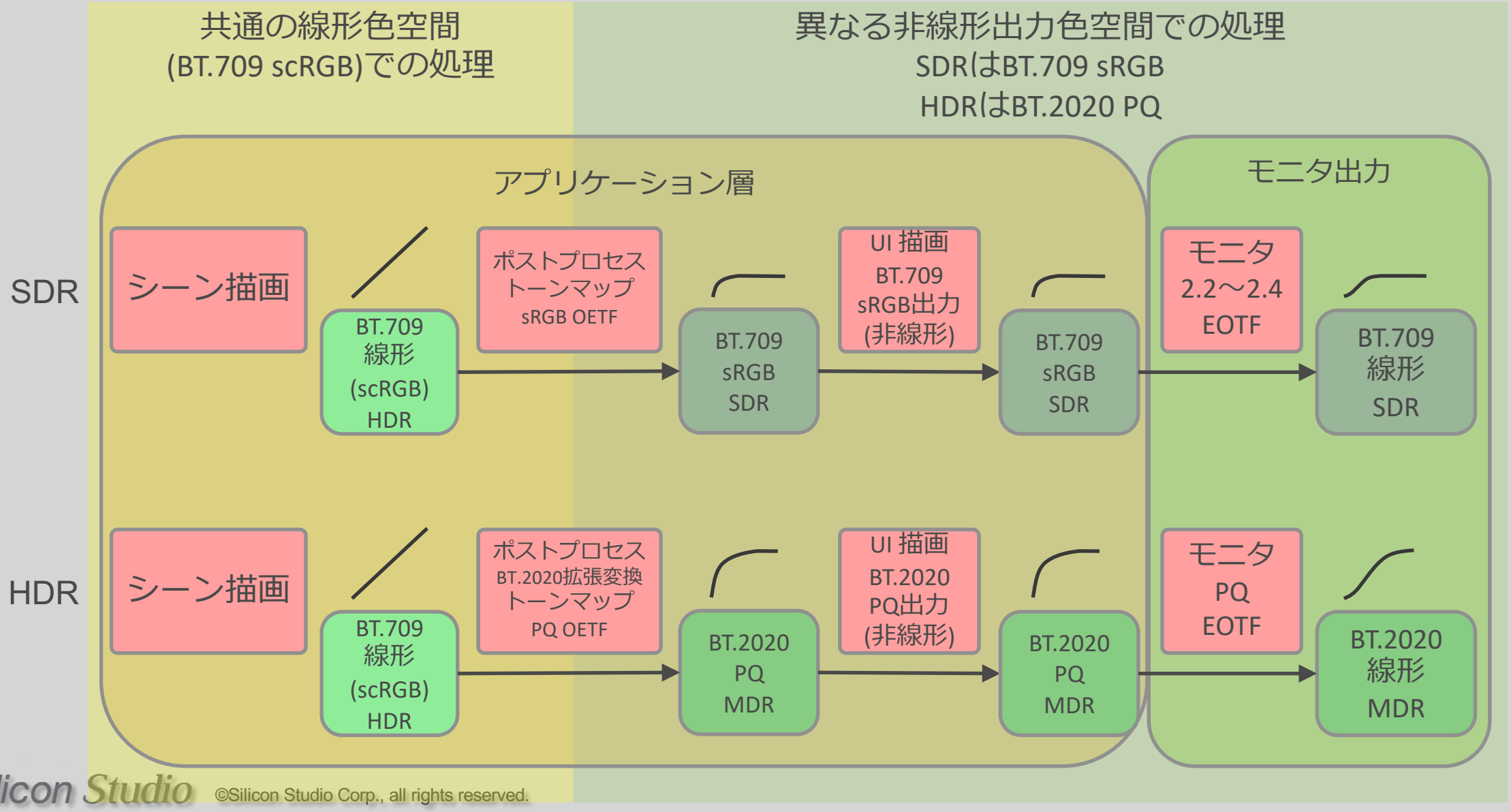

©Silicon Studio Corp., all rights reserved.

# 即時OETF

- HDR10等
  - BT.2020 PQ非線形色空間での書き出し
    - RGB10_A2/RGBA16_UNORMフォーマットに出力
  - scRGB 線形色空間での書き出し
    - RGBA16_FLOATフォーマットに出力
- HLG
  - 現状ではリアルタイムコンテンツにおける対応無し
    - 対応する場合はRGB10_A2/RGBA16_UNORMに出力
- Dolby Vision
  - RGBA8_UNORMフォーマットに出力
    - 内部データとしては12-bit YCbCr 4:2:2
- etc.

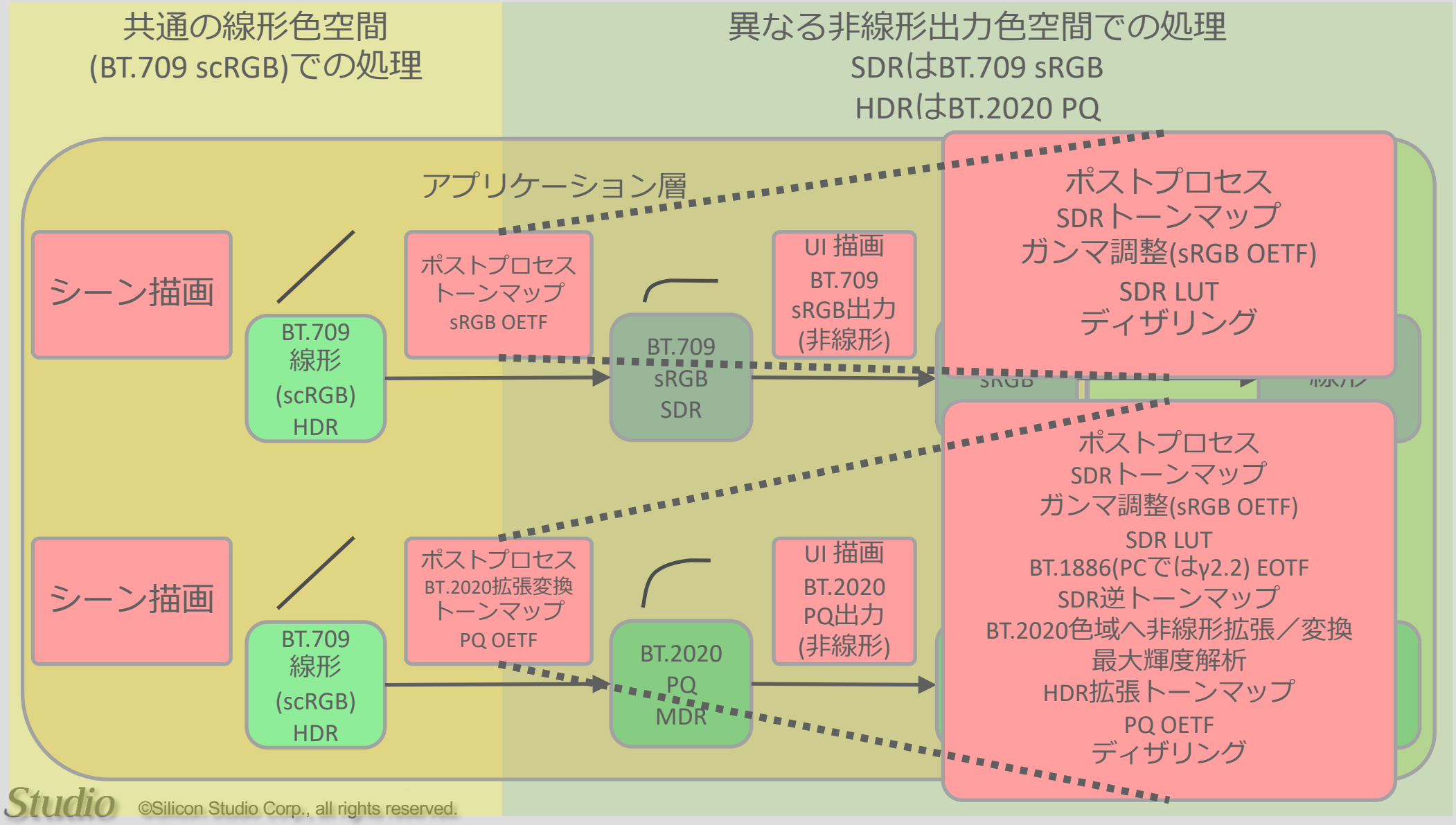
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 再掲：BT.709ベース(UI描画は出力色空間別)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 詳細：BT.709ベース(UI描画は出力色空間別)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 即時OETFまとめ

- パフォーマンス重視
- チェッカーボードレンダリングが効率的
  - ポストプロセスに制限あり
- UI描画の色空間が異なる
  - HDR時のUIブレンドが不正確

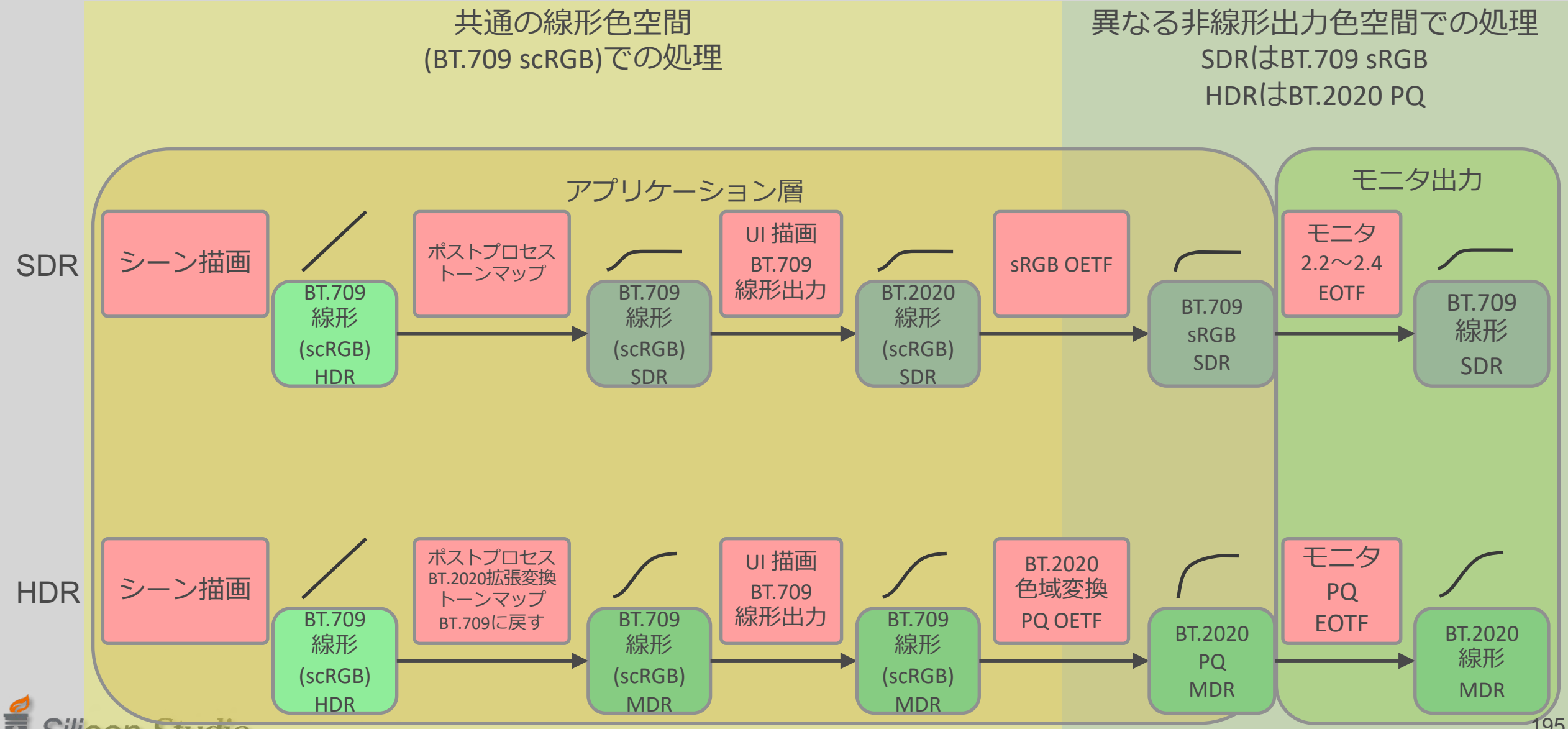

©Silicon Studio Corp., all rights reserved.

# アプリ側遅延OETF

- HDR10等
  - BT.2020 PQ非線形色空間での書き出し
    - RGB10_A2/RGBA16_UNORMフォーマットに出力
  - scRGB 線形色空間での書き出し
    - RGBA16_FLOATフォーマットに出力
- HLG
  - 現状ではリアルタイムコンテンツにおける対応無し
    - 対応する場合はRGB10_A2/RGBA16_UNORMに出力
- Dolby Vision
  - RGBA8_UNORMフォーマットに出力
    - 内部データとしては12-bit YCbCr 4:2:2
- etc.

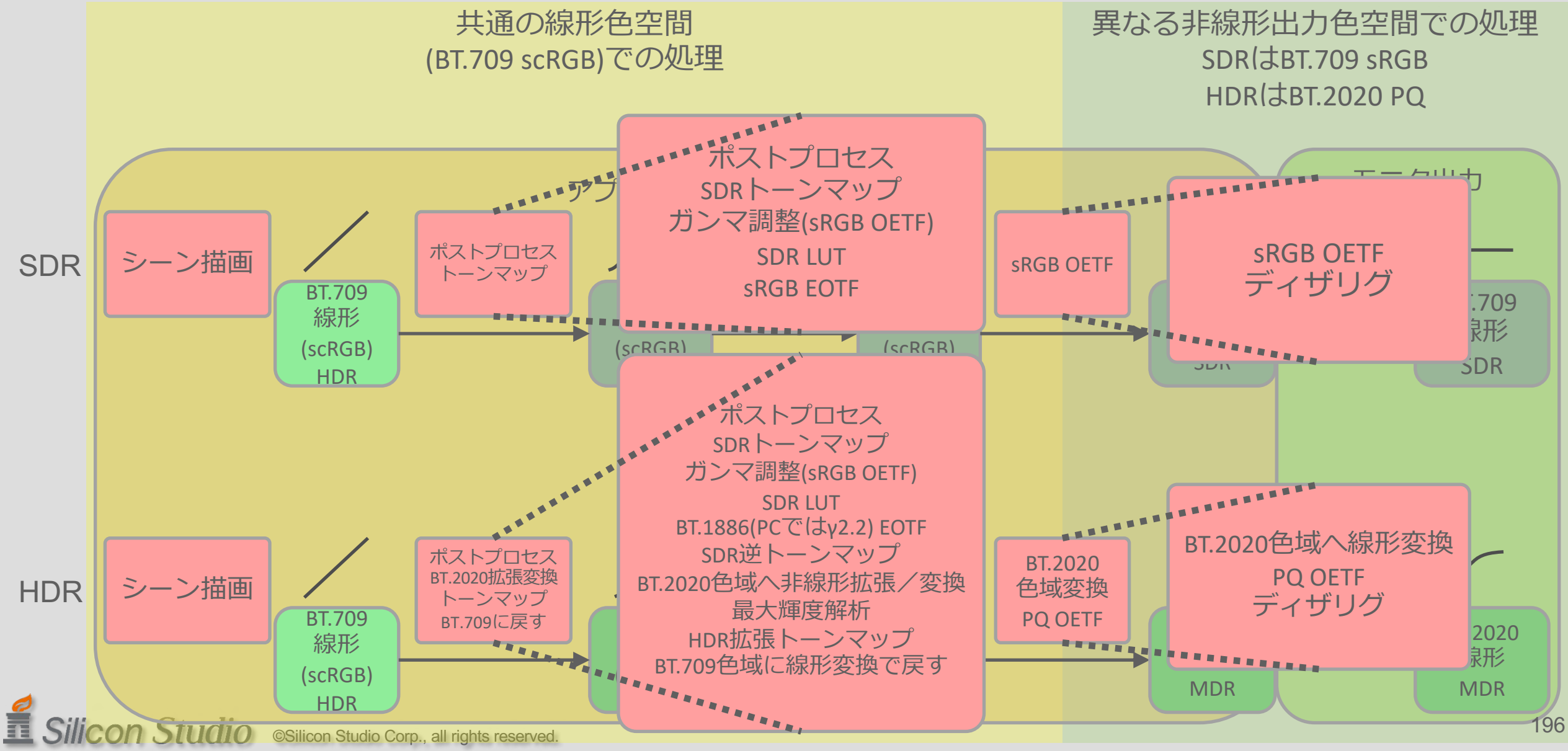

©Silicon Studio Corp., all rights reserved.

# 再掲：BT.709ベース(UI描画もBT.709線形)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 詳細：BT.709ベース(UI描画もBT.709線形)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# アプリ側遅延OETFまとめ

- SDRとの互換性重視
  - フローを通して色空間が同じ
  - UIブレンドが正しい
- パフォーマンスはやや劣る
  - 最後に変換パスが必要
  - チェッカーボードレンダリングが非効率

- Dolby Visionでは必ずこのフロー
  - 必ず描画完了後にエンコードが必要になるため

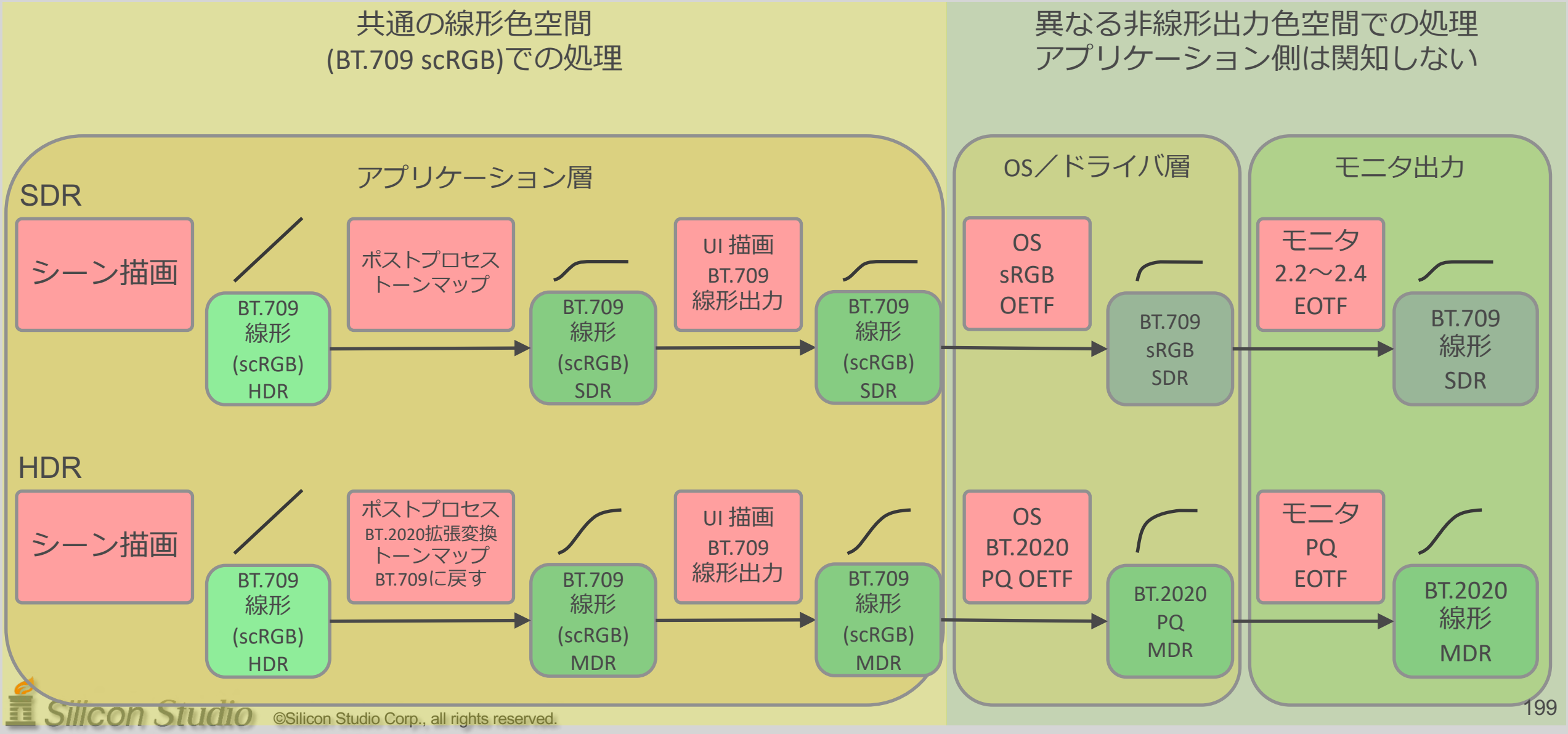
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OS側遅延OETF(scRGB出力)

- HDR10等
  - BT.2020 PQ非線形色空間での書き出し
    - RGB10_A2/RGBA16_UNORMフォーマットに出力
  - scRGB 線形色空間での書き出し
    - RGBA16_FLOATフォーマットに出力
- HLG
  - 現状ではリアルタイムコンテンツにおける対応無し
    - 対応する場合はRGB10_A2/RGBA16_UNORMに出力
- Dolby Vision
  - RGBA8_UNORMフォーマットに出力
    - 内部データとしては12-bit YCbCr 4:2:2
- etc.

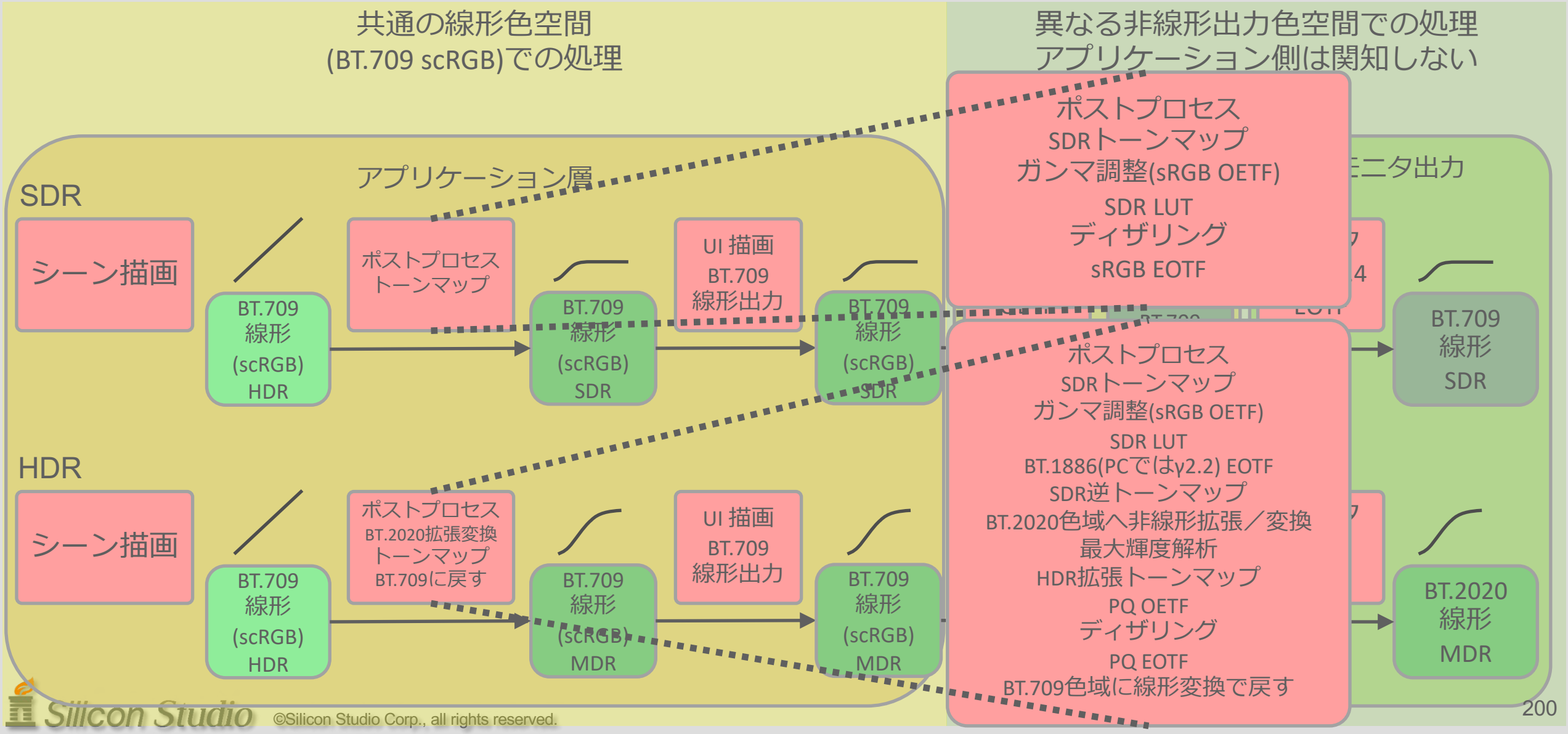
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 再掲：BT.709ベース(すべてscRGBに出力)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 詳細：BT.709ベース(すべてscRGBに出力)




©Silicon Studio Corp., all rights reserved.

# OS側遅延OETF(scRGB出力)まとめ

- 全体的にやや非効率
  - OS／ドライバ層での変換が入る
  - 互換性を重視すると無駄なパスが増える
  - チェッカーボードレンダリングが非効率
- SDRとの互換性が高い
  - フローを通して色空間が同じ
  - UIブレンドは基本的には正しい
    - 色域拡張後のscRGBで乗算ブレンド等を行うと不適切になる可能性あり
- NVAPIでは必ずこのフロー

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 既存のSDR出力との互換性を重視したフロー まとめ

- 互換性保持のために必要な処理が多い
- SDR LUTやディザリング等
  - 重要性やコストパフォーマンス如何で不採用も検討


©Silicon Studio Corp., all rights reserved.

# 前提条件を変えたフローも考えられる

- 例えば新規開発でSDRフローも刷新できる場合
  - 仮に条件として
    - レンダリングはBT.2020ベース
    - LUTもHDR空間
      - LUT用の何らかの効率的な色空間で適用
    - SDRのガンマ補正を利用したコントラスト調整は行わない
      - LUTやトーンマップを利用
    - UI描画はBT.2020の素材を出力に合わせて逐次変換
      - 色域圧縮マッピング


©Silicon Studio Corp., all rights reserved.

# 前提条件を変えたSDR色空間フロー

- HDR LUT入力色空間へ変換
- HDR LUT適用で線形に戻る
- BT.709色域への非線形圧縮と変換
- 最大輝度解析
- SDRトーンマップ
- sRGB OETF
- ディザリング

- HDR出力を含む汎用フローのため従来のSDR出力より複雑


©Silicon Studio Corp., all rights reserved.

# 前提条件を変えたHDR色空間フロー

- HDR LUT入力色空間へ変換
- HDR LUT適用で線形に戻る
- 最大輝度解析
- HDR拡張トーンマップ
- sRGB OETF
- BT.1886(PCではγ2.2) EOTF
- PQ OETF
- ディザリング

- 過去のSDRとの互換性重視フローよりも全体的にシンプル

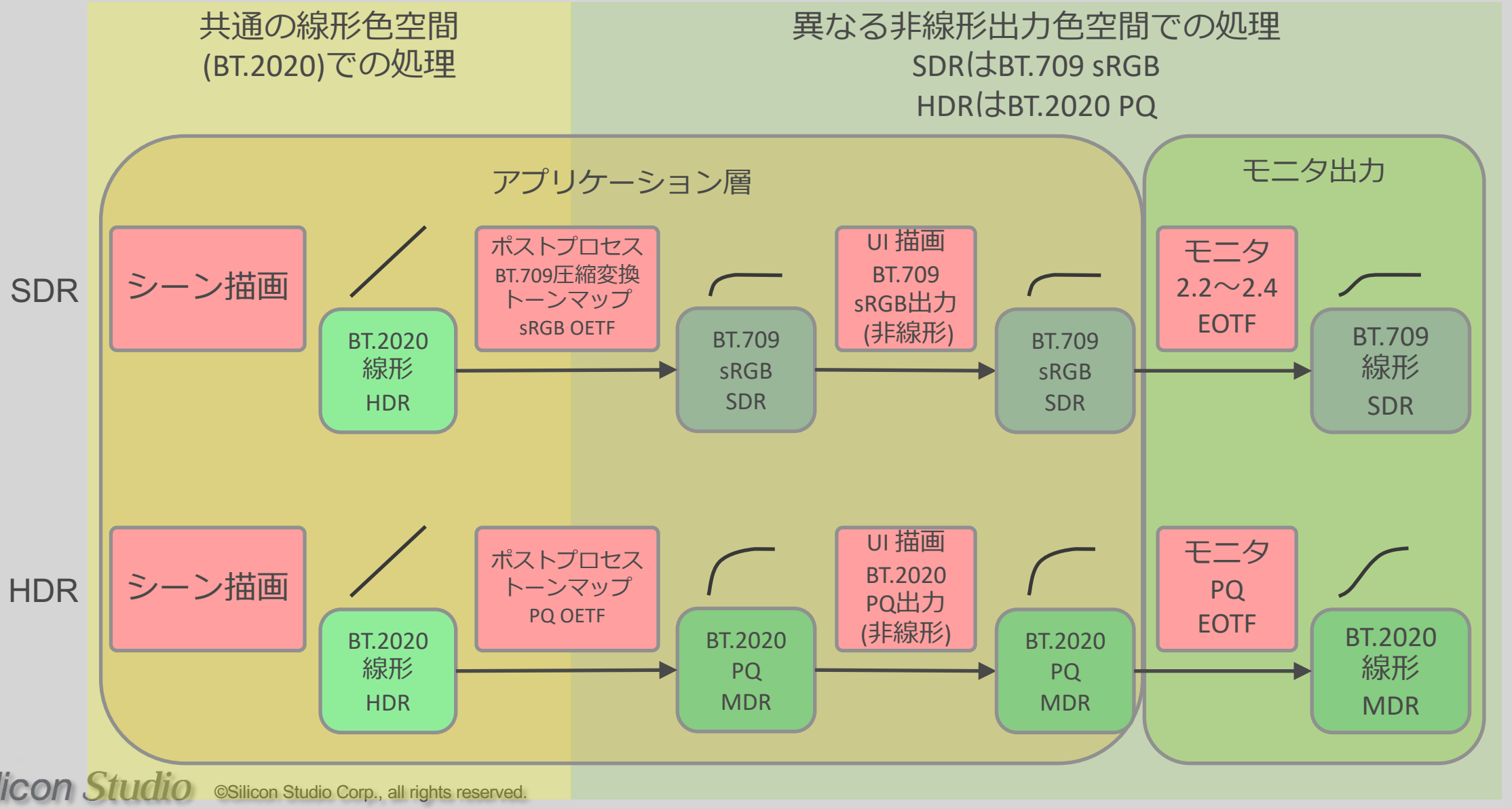
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 即時OETF

- HDR10等
  - BT.2020 PQ 非線形色空間での書き出し
    - RGB10_A2/RGBA16_UNORMフォーマットに出力
  - scRGB 線形色空間での書き出し
    - RGBA16_FLOATフォーマットに出力
- HLG
  - 現状ではリアルタイムコンテンツにおける対応無し
    - 対応する場合はRGB10_A2/RGBA16_UNORMに出力
- Dolby Vision
  - RGBA8_UNORMフォーマットに出力
    - 内部データとしては12-bit YCbCr 4:2:2
- etc.

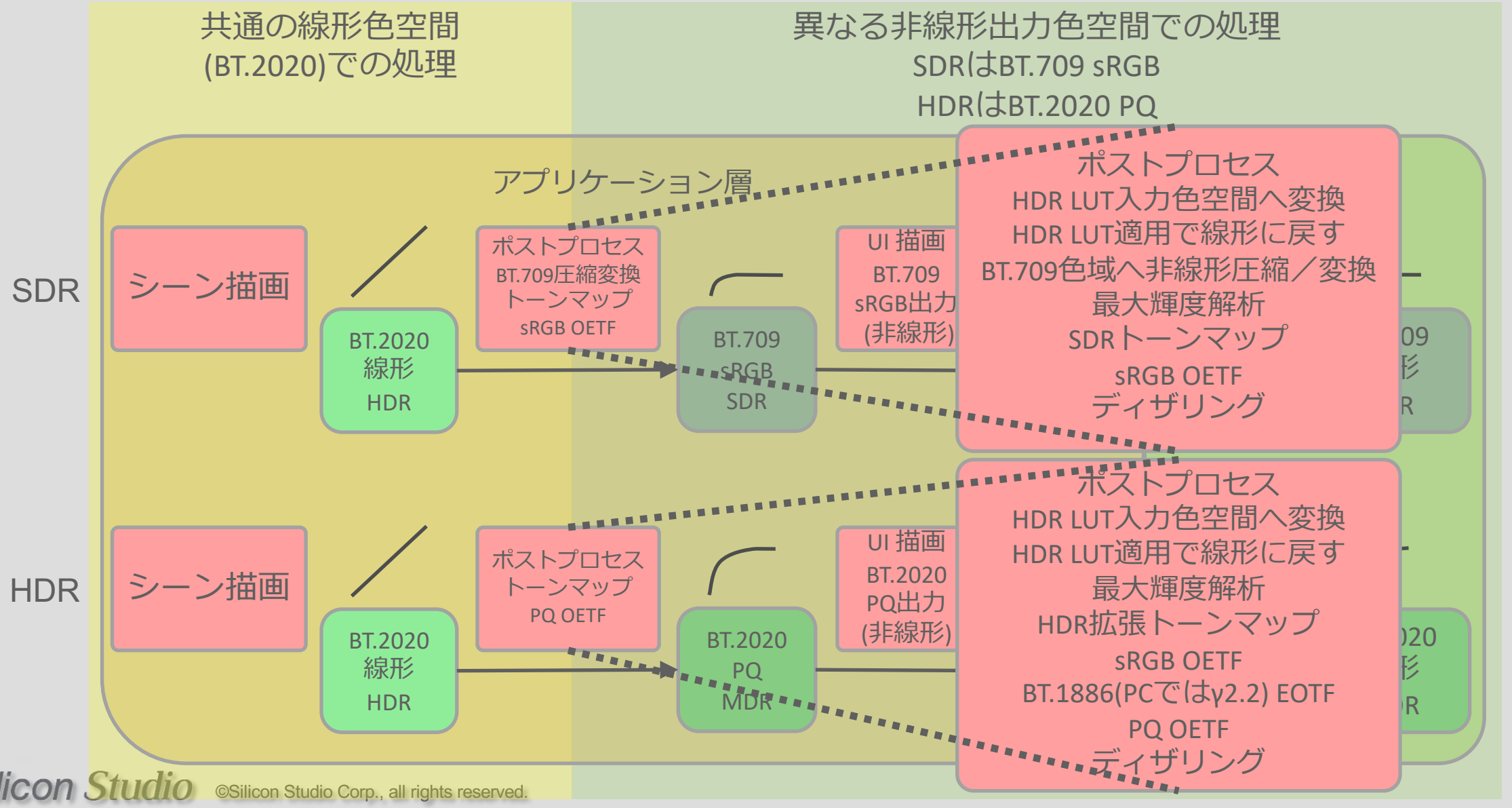
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 再掲：BT.2020ベース(UI描画は出力色空間別)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 詳細：BT.2020ベース(UI描画は出力色空間別)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 即時OETFまとめ

- パフォーマンス重視
- チェッカーボードレンダリングが効率的
  - ポストプロセスに制限あり
- UI描画の色空間が異なる
  - HDR時のUIブレンドが不正確

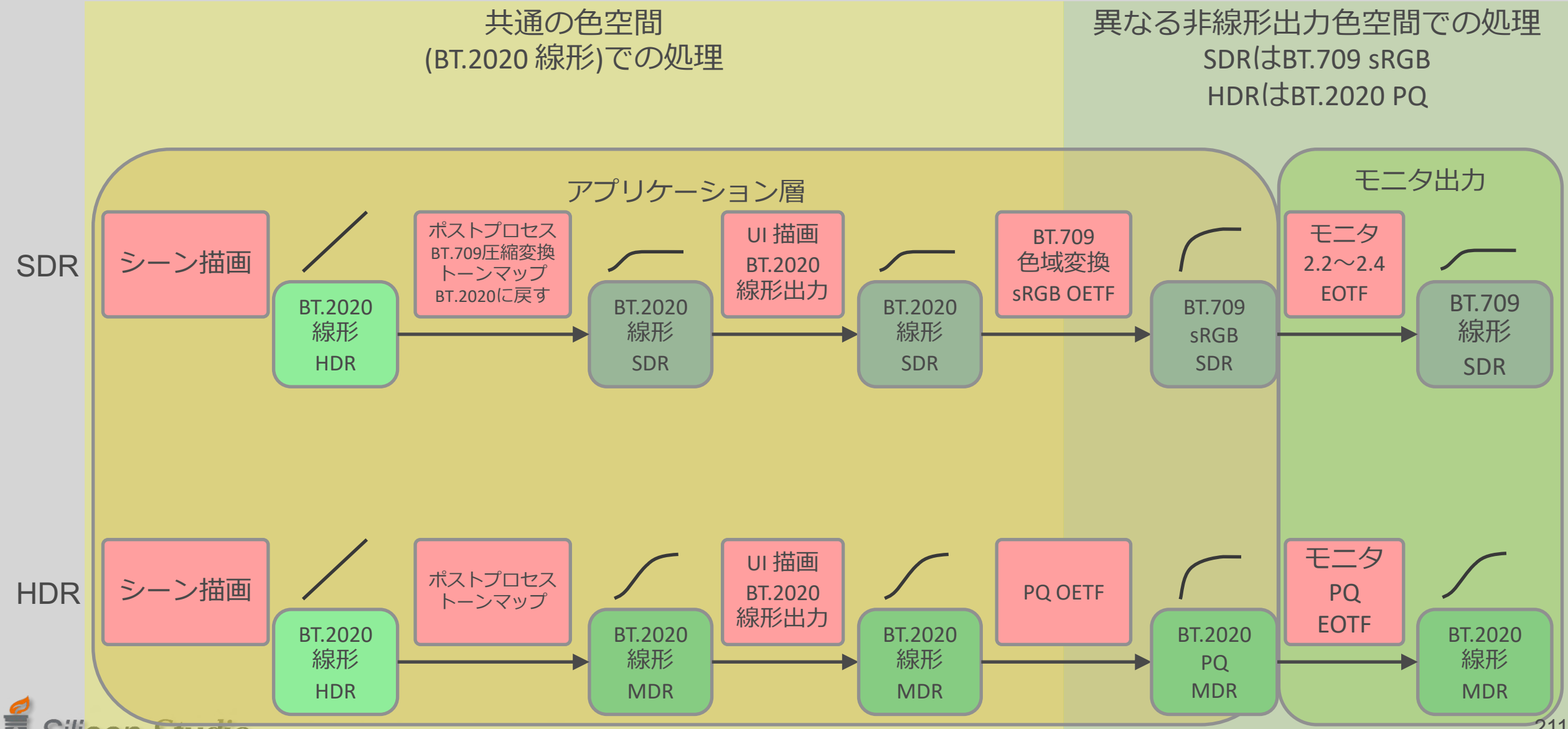

©Silicon Studio Corp., all rights reserved.

# アプリ側遅延OETF

- HDR10等
  - BT.2020 PQ 非線形色空間での書き出し
    - RGB10_A2/RGBA16_UNORMフォーマットに出力
  - scRGB 線形色空間での書き出し
    - RGBA16_FLOATフォーマットに出力
- HLG
  - 現状ではリアルタイムコンテンツにおける対応無し
    - 対応する場合はRGB10_A2/RGBA16_UNORMに出力
- Dolby Vision
  - RGBA8_UNORMフォーマットに出力
    - 内部データとしては12-bit YCbCr 4:2:2
- etc.

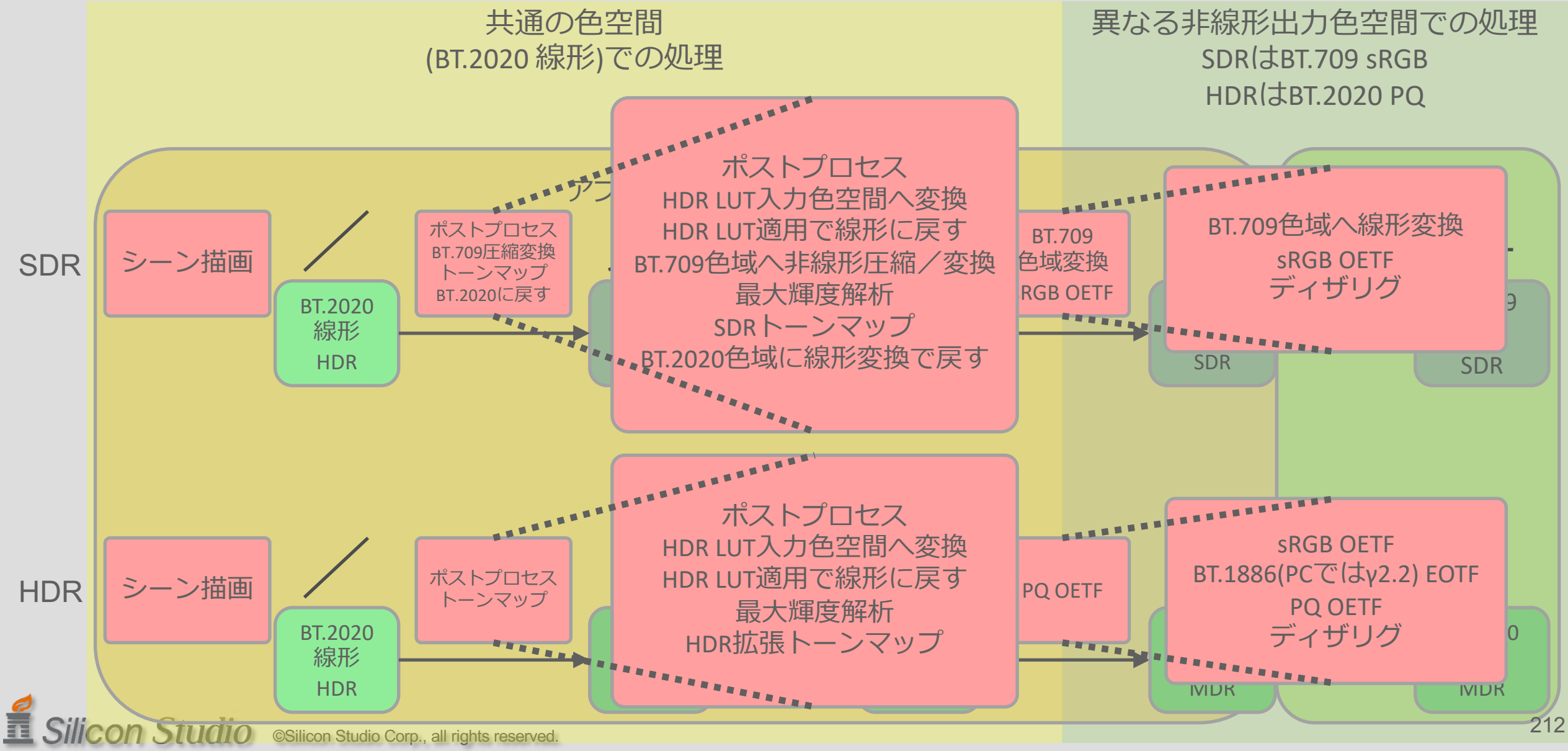
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 再掲：BT.2020ベース(UI描画もBT.2020線形)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 詳細：BT.2020ベース(UI描画もBT.2020線形)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# アプリ側遅延OETFまとめ

- SDRとの互換性重視
  - フローを通して色空間が同じ
  - UIブレンドが正しい
- パフォーマンスはやや劣る
  - 最後に変換パスが必要
  - チェッカーボードレンダリングが非効率

- Dolby Visionでは必ずこのフロー
  - 必ず描画完了後にエンコードが必要になるため

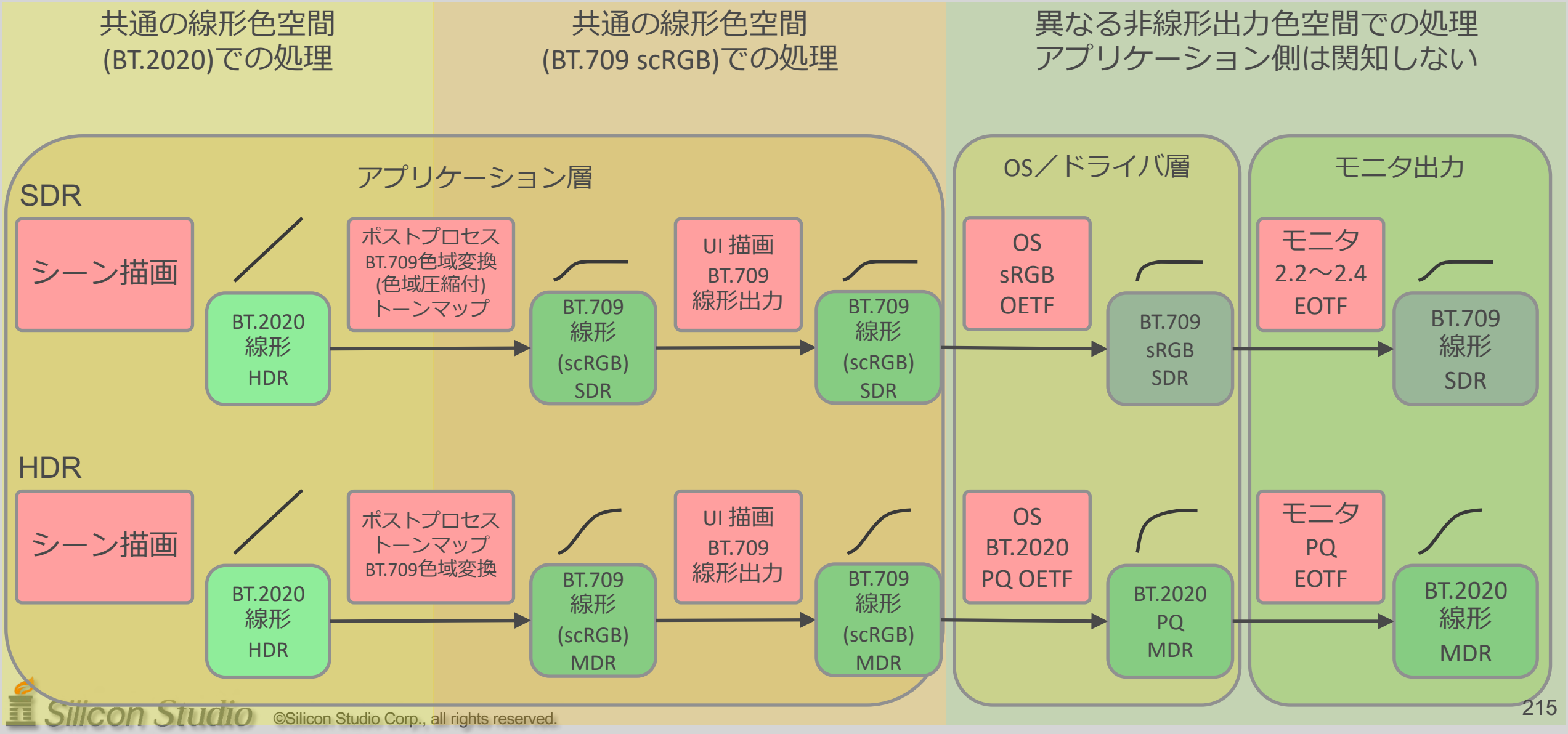
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# OS側遅延OETF(scRGB出力)

- HDR10等
  - BT.2020 PQ 非線形色空間での書き出し
    - RGB10_A2/RGBA16_UNORMフォーマットに出力
  - scRGB 線形色空間での書き出し
    - RGBA16_FLOATフォーマットに出力
- HLG
  - 現状ではリアルタイムコンテンツにおける対応無し
    - 対応する場合はRGB10_A2/RGBA16_UNORMに出力
- Dolby Vision
  - RGBA8_UNORMフォーマットに出力
    - 内部データとしては12-bit YCbCr 4:2:2
- etc.

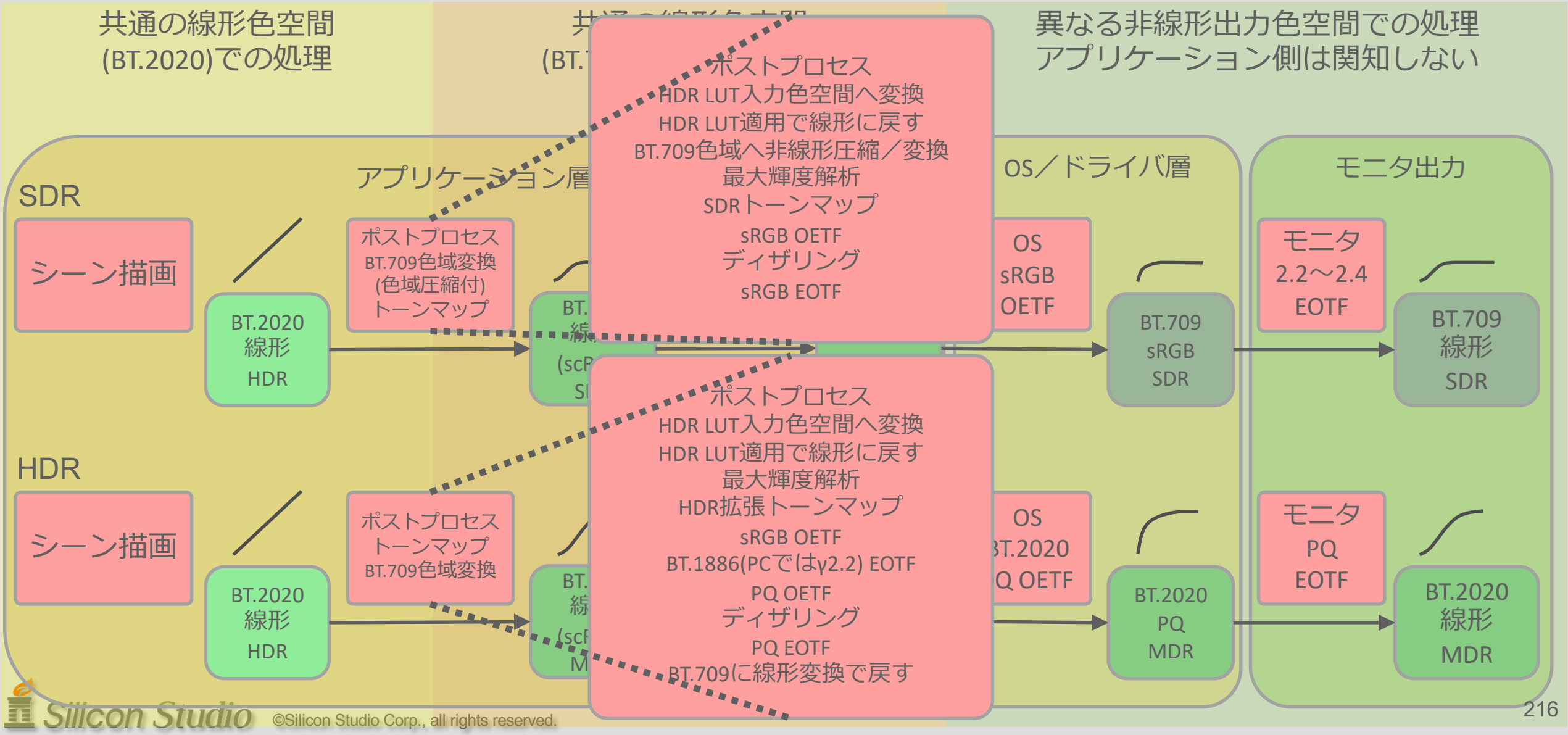
Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 再掲：BT.2020ベース(UI描画はscRGB)




Silicon Studio ©Silicon Studio Corp., all rights reserved.

# 詳細：BT.2020ベース(UI描画はscRGB)




©Silicon Studio Corp., all rights reserved.

# OS側遅延OETF(scRGB出力)まとめ

- 全体的にやや非効率
  - OS／ドライバ層での変換が入る
  - 互換性を重視すると無駄なパスが増える
  - チェッカーボードレンダリングが非効率
- SDRとの互換性が高い
  - フローを通して色空間が同じ
  - UIブレンドは基本的には正しい
    - 広色域のためscRGBで乗算ブレンド等を行うと不適切になる可能性あり
- NVAPIでは必ずこのフロー

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# まとめ

本セッションの目的
色空間
前提として押さえておきたいこと
基本的なレンダリングフローの検討
その他の考慮すべきこと
互換性重視レンダリングフローの構築
まとめ

Silicon Studio ©Silicon Studio Corp. all rights reserved!

# SDRと互換性を保つレンダリングフロー

- さまざまな要素の検討が必要
  - SDR出力時に何の処理がどの色空間で行われているか
  - 順番等を入れ替えた場合の影響の考慮
  - 異なる色空間での絵作りやパラメタの互換性をどう保つか
- とにかく色空間の遷移を正確に把握すること
  - 論理的に考察すると意外と複雑になる…


©Silicon Studio Corp., all rights reserved.

# SDRと互換性を保つレンダリングフロー

- とりあえず最低限の対応なら…
  - アプリ側の遅延OETFだけでも良い
    - パフォーマンス重視の即時OETFもあると選択肢は増える
  - OS層による遅延OETF(アプリ側scRGB出力)は
    - 互換性その他を考慮すると無駄が多くなる
    - 多くのケースでは他の選択肢があるため必須ではない

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# SDRと互換性を保つレンダリングフロー

- 必ずしも紹介した方法が正解ではない
  - SDRレンダリング時の条件に依存
  - 重要度の低い要素の省略
  - 簡略化できる部分の検討
- 重要なのはフロー構築のための考え方
  - 各々に最適なフローを考察してみてください
  - 紹介した内容がヒントになれば幸いです

Silicon Studio ©Silicon Studio Corp., all rights reserved.

# References


©Silicon Studio Corp., all rights reserved!

# References


- Recommendation ITU-R BT.601 “Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios.” http://www.itu.int/rec/R-REC-BT.601/, 2011.
- Recommendation ITU-R BT.709 “Parameter values for the HDTV standards for production and international programme exchange.” http://www.itu.int/rec/R-REC-BT.709/, 2015.
- Recommendation ITU-R BT.1886 “Reference electro-optical transfer function for flat panel displays used in HDTV studio production.” http://www.itu.int/rec/R-REC-BT.1886/, 2011.
- Recommendation ITU-R BT.2020 “Parameter values for ultra-high definition television systems for production and international programme exchange.” http://www.itu.int/rec/R-REC-BT.2020/, 2015.
- Recommendation ITU-R BT.2100 “Image parameter values for high dynamic range television for use in production and international programme exchange.” http://www.itu.int/rec/R-REC-BT.2100/, 2016.
- Report ITU-R BT.2390 “High dynamic range television for production and international programme exchange.” https://www.itu.int/pub/R-REP-BT.2390/, 2016.
- BBC Research & Development, “Perceptual Quantiser (PQ) to Hybrid Log-Gamma (HLG) Transcoding.” http://www.bbc.co.uk/rd/sites/50335ff370b5c262af000004/assets/592eea8006d63e5e5200f90d/BBC_HDRTV_PQ_HLG_Transcode_v2.pdf, HDR-TV, 2017.
- ARIB STD-B67 “ESSENTIAL PARAMETER VALUES FOR THE EXTENDED IMAGE DYNAMIC RANGE TELEVISION (EIDRTV) SYSTEM FOR PROGRAMME PRODUCTION ARIB STANDARD.” http://www.arib.or.jp/tyosakenkyu/kikaku_hoso/hoso_std-b067.html, 2015.
- MSDN, “High Dynamic Range and Wide Color Gamut” https://msdn.microsoft.com/en-us/library/windows/desktop/mt742103(v=vs.85).aspx
- NVIDIA GameWorks, “High Dynamic Range Display Development.” https://developer.nvidia.com/high-dynamic-range-display-development



©Silicon Studio Corp., all rights reserved.

# References


- [Kawase16] Kawase, M. “シリコンスタジオによるHDR出力対応の理論と実践” http://www.siliconstudio.co.jp/rd/presentations/#CEDEC2016 Computer Entertainment Developers Conference 2016.
- [Uchimura17] Uchimura, H. “HDR 理論と実践” https://www.slideshare.net/nikuque/hdr-theory-and-practicce-jp Computer Entertainment Developers Conference 2017.
- [Gotanda07] Gotanda, Y. “レンダリストのためのカメラ（光学）理論とポストエフェクト” http://research.tri-ace.com/ CESA Developers Conference 2007.
- [Lottes16] Lottes, T. “Advanced Techniques and Optimization of HDR VDR Color Pipelines.” https://32ipi028l5q82yhj72224m8j-wpengine.netdna-ssl.com/wp-content/uploads/2016/03/GdcVdrLottes.pdf, Game Developers Conference 2016.



©Silicon Studio Corp., all rights reserved.